

diafan.CMS version 5.3
<http://cms.diafan.ru/>



Полное руководство для разработчиков

Содержание

- [Введение](#)
 - [Технические требования к хостингу](#)
 - [Как установить diafan.CMS](#)
 - [Как зарегистрировать diafan.CMS](#)
 - [Файловая структура diafan.CMS](#)
 - [Вход в панель администрирования diafan.CMS](#)
 - [Первое знакомство с панелью управления сайтом](#)
 - [Панель редактирования страниц сайта \(визуальный редактор\)](#)
 - [Конфигурация сайта config.php](#)
- [Интеграция дизайна, шаблоны и шаблонные функции](#)
 - [Как подключить diafan.CMS к дизайну](#)
 - [Шаблон сайта](#)
 - [Шаблоны модулей](#)
 - [Шаблонные функции и теги](#)
 - [Шаблонные функции модулей](#)
- [Системные модули](#)
 - [Страницы сайта](#)
 - [Меню на сайте](#)
 - [Пользователи сайта](#)
 - [Обновление CMS](#)
 - [Модули и БД](#)
 - [Параметры сайта](#)
 - [Языки сайта](#)
 - [Корзина](#)
- [Модули](#)
 - [Новости](#)
 - [Магазин](#)
 - [Фотогалерея](#)
 - [Статьи](#)
 - [Баннеры](#)
 - [Файловый архив](#)
 - [Объявления](#)
 - [Теги](#)
 - [Поиск](#)
 - [Вопрос-Ответ](#)
 - [Обратная связь](#)
 - [Форум](#)
 - [Опросы](#)
 - [Рассылки](#)
 - [Карта сайта](#)
 - [Ошибка на сайте](#)
 - [On-line консультант](#)
 - [Web Effector](#)
 - [Файловый менеджер](#)
- [Надмодули](#)

- [Комментарии](#)
- [Рейтинг](#)
- [Перелинковка](#)
- [Панель быстрого редактирования](#)
- [Изображения](#)
- [Прикрепленные файлы](#)
- [Постраничная навигация](#)
- [Captcha](#)
- [Разработчикам](#)
 - [Основные принципы diafan.CMS](#)
 - [Архитектура diafan.CMS](#)
 - [Структура базы данных](#)
 - [Устройство модуля](#)
 - [Контроллер](#)
 - [Модель](#)
 - [Шаблон](#)
 - [Обработка Ajax-запросов](#)
 - [Подключение](#)
 - [Инсталляция](#)
 - [Прямое обращение к файлам модуля](#)
 - [Взаимодействие с другими модулями](#)
 - [Подключение модуля](#)
 - [Административная часть сайта](#)
 - [Основные действия](#)
 - [Администрирование модуля](#)
 - [Настройки модуля](#)
 - [Подключение модуля](#)
 - [Подключения \(API\)](#)
 - [База данных](#)
 - [Валидация данных](#)
 - [Изображения](#)
 - [Кастомизация](#)
 - [Конфигурация сайта](#)
 - [Кэширование](#)
 - [Маршрутизатор](#)
 - [Общие функции](#)
 - [Подключение шаблонов](#)
 - [Пользователи](#)
 - [Почта](#)
 - [Режим разработки](#)
 - [Сессии](#)
 - [Сжатие страниц](#)
 - [Установка модулей](#)
 - [Файлы](#)
 - [SMS](#)
 - [Список внешних плагинов](#)
- [Практические примеры](#)
 - [Как создать свой модуль](#)
 - [Как добавить шаблонный тег](#)

- [Как добавить новое поле в модуль](#)
- [Как получить свободные данные из БД и вывести их](#)
- [Как вставить сторонний PHP-код](#)
- [Как сделать зависимые Ajax-списки](#)
- [Как в модуле использовать другой модуль](#)
- [Переопределение функционала в административной части модуля](#)
- [Кастомизация функций](#)
- [Изменение административной части](#)
- [Как в поиске по товарам заменить чекбоксы на список](#)
- [Как вывести дополнительную характеристику товара в любом месте](#)
- [Как вывести список товаров в корзине в шаблонном теге](#)
- [Как добавить виджет комментариев из социальной сети](#)
- [Справочник шаблонных тегов](#)

Введение

- [Терминология](#)

Система управления сайтами diafan.CMS предназначена для создания и управления интернет-проектами.

diafan.CMS позволяет легко и быстро подключать любой дизайн, удобно создавать структуру сайта неограниченной вложенности, наполнять сайт различной информацией любому пользователю, обладающему элементарными навыками по работе с офисными приложениями.

Терминология

Пользовательская часть сайта – общедоступная для всех пользователей часть сайта, доступна по адресу <http://site.ru/>.

Административная часть сайта – часть сайта, предназначенная для редактирования информации пользовательской части. Доступна только для администратора, после ввода логина и пароля. Адрес административной части задается администратором при установке diafan.CMS. Например: <http://www.site.ru/admin/> или <http://www.site.ru/adminka/>.

Шаблонный тег – команда для вывода информации в пользовательской части сайта. Шаблонный тег запускает шаблонную функцию.

Шаблон – файл оформления пользовательской части сайта. Это текстовый файл HTML-кода со вставками шаблонных тегов.

Шаблонная функция – функция на PHP, выводящая контент в пользовательской части сайта. Запускается с помощью шаблонного тега.

Модуль – часть сайта, работающая по определенному алгоритму (Например, модуль «Новости» – вывод группы новостей с датой, анонсом и заголовком). Модуль прикрепляется к одной или нескольким страницам сайта в административной части сайта и выполняет некоторые действия по формированию страницы в пользовательской части сайта.

Подключение – набор дополнительных PHP-функций, подключаемых для осуществления определенных целей (работа с файлами, работа с базой данных, работа с изображениями).

Плагины – функционал сторонних разработчиков, используемый для выполнения определенных задач.

Страница сайта – в пользовательской части часть сайта с уникальным адресом <http://site.ru/stranica/>, оформленная шаблоном дизайна, с заголовком и текстовой информацией. В административной части страницы сайта редактируются в основном модуле «Страницы сайта», к странице может быть подключен один модуль, несколько подключений и шаблонных функций.

Технические требования к хостингу

- [Установка на localhost](#)

Для работы diafan.CMS подойдет практически любой российский платный хостинг на UNIX-платформе. На сервере должно быть установлено:

- Apache с установленным mod_rewrite
- PHP >= 5.2X
- Библиотеки GD
- MySQL 5.XX
- В настройках Apache должна быть включена директива AllowOverride со значением «All» или «Options FileInfo AutoConfig», и при этом директива AccessFileName имеет значение .htaccess
- Поддержка сокетов и деректива allow_url_fopen
- Опционально --enable-ftp для автообновления

Установка на localhost

Возможна работа diafan.CMS на локальном хосте. Для этого подойдет любая программная оболочка-эмулятор, набор Apache+PHP+MySQL. В частности, [Джентельменский набор](#). Чтобы diafan.CMS не требовала лицензию, необходимо, чтобы локальный сервер назывался localhost, diafan, mysite или любое иное имя без точки. Последовательность инсталляции на локальный сервер идентична [инсталляции на хостинг](#).

Как установить diafan.CMS

- [Начало установки](#)

Внимание! [Смотрите видеоурок на тему установки diafan.CMS](#)

Начало установки

1. Скопируйте содержимое архива diafan.CMS.zip (скачать можно в [личном кабинете](#)) в корневую директорию своего сайта.
2. Наберите в адресной строке браузера адрес своего сайта.
3. Следуйте инструкциям.

На первом шаге будут проверены настройки сервера. Если что-то не соответствует требованиям diafan.CMS, установщик сообщит об этом.

Установка

- Проверка системы
- Лицензия
- Параметры
- Администратор сайта
- Модули
- Окончание

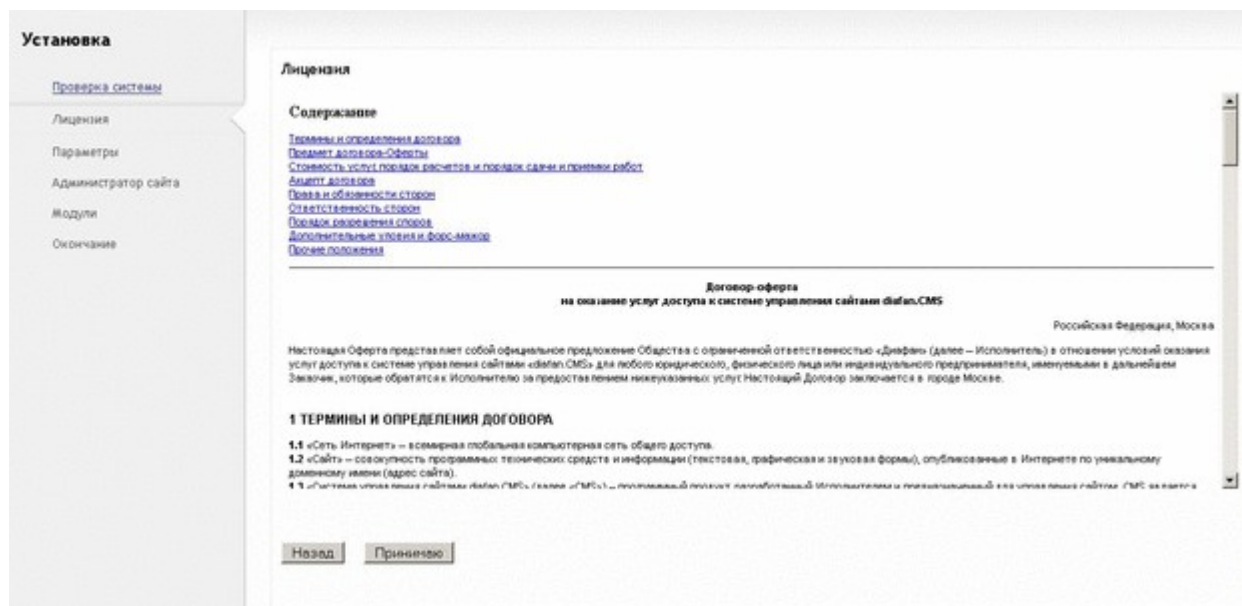
Проверка системы

Версия PHP >= 5.0	да
Поддержка MySQL	да
Поддержка сокетов	да
/userfiles/	да
/cache/	да
/installation/	да
Файл config.php	да
Кэш index.html	да
language/ru.php	да

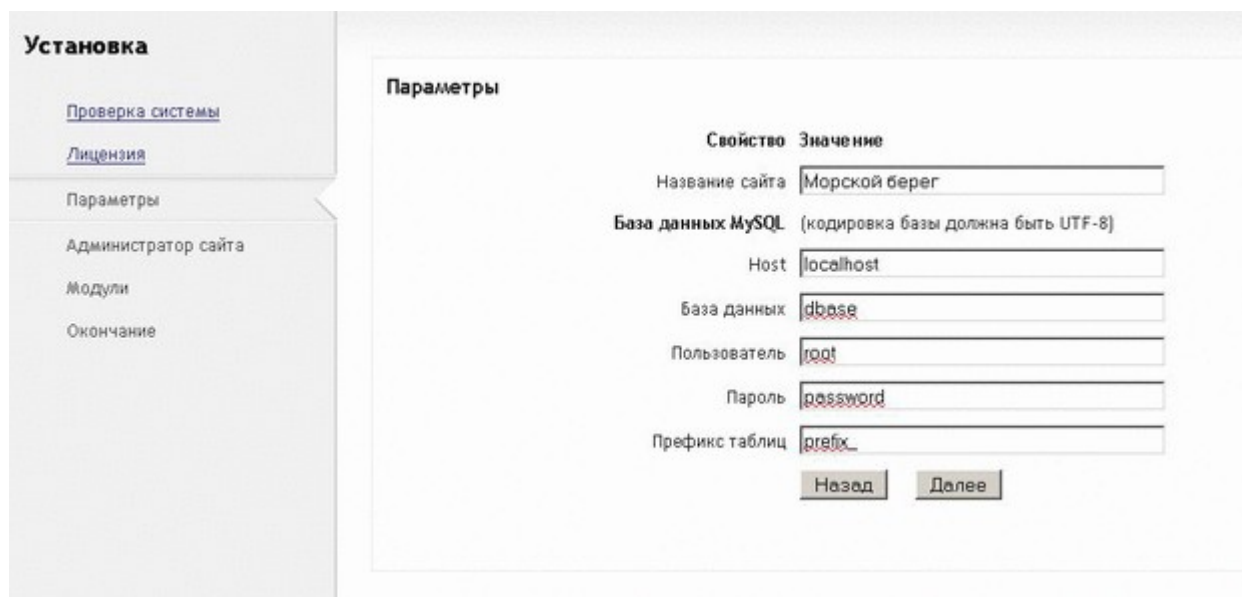
Проверить снова

Далее

Далее следует принять договор-оферту.



На третьем шаге необходимо ввести параметры сайта и базы данных сервера. Если будут введены неверные параметры базы данных, установщик сообщит об этом. Продолжение установки не будет возможно.



Далее следует создать главного администратора системы. Нужно заполнить все поля. Старайтесь не использовать слишком простые пароли.

Установка

[Проверка системы](#)
[Лицензия](#)
[Параметры](#)
Администратор сайта
Модули
Окончание

Администратор сайта

Свойство	Значение
ФИО	Иванов Иван
Логин	login
Пароль	password
E-mail адрес	mail@mail.ru
Папка для панели администрирования	admin

Назад
Далее

Затем необходимо указать дополнительные модули, которые необходимо установить.

Установка

[Проверка системы](#)
[Лицензия](#)
[Параметры](#)
[Администратор сайта](#)
Модули
Окончание

Модули

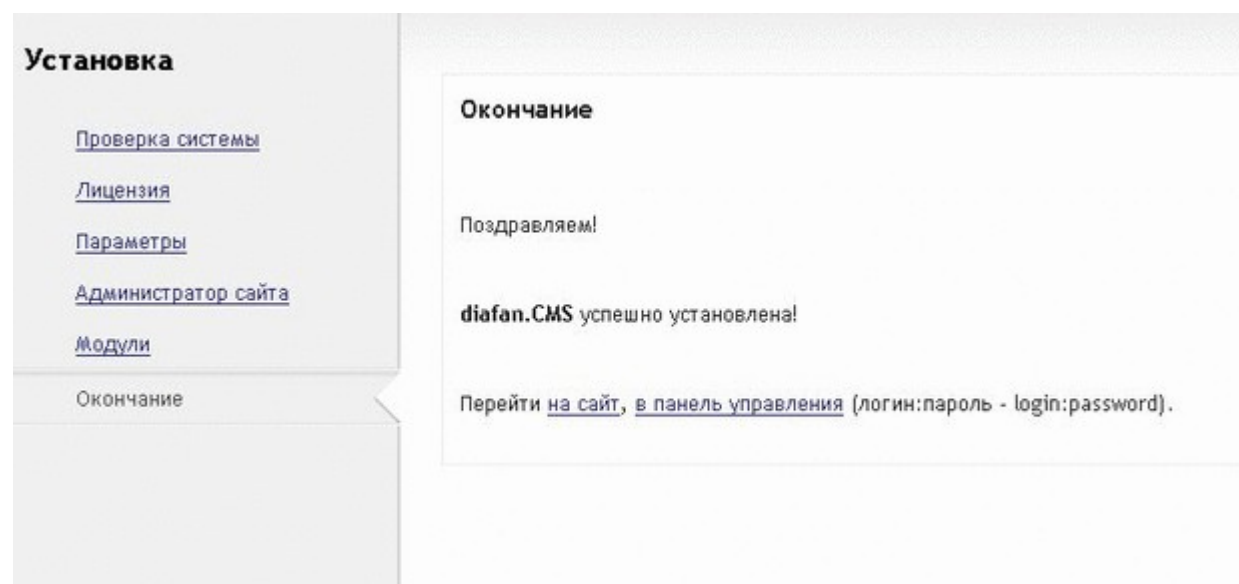
Какие дополнительные модули установить?

- ☒ Пользователи сайта
- ☒ Личные сообщения
- ☒ Новости
- ☒ Статьи
- ☒ Форма обратной связи
- ☒ Вопрос-Ответ
- ☒ Магазин
- ☒ Файлы
- ☒ Поиск
- ☒ Баннеры
- ☒ Комментарии
- ☒ Рейтинг
- ☒ Теги
- ☒ Фотогалерея
- ☒ Опросы
- ☒ Рассылки
- ☒ Форум
- ☒ Карта сайта

☐ Заполнить сайт примерами из демо-версии «Бумажный зоопарк» (займет несколько минут)
☐ Создать две языковые версии сайта (Русский/English)

Назад
Далее

Последний шаг – окончание установки. В случае, если домен, на который установлена diafan.CMS не зарегистрирован в [личном кабинете](#), следует это [сделать](#).



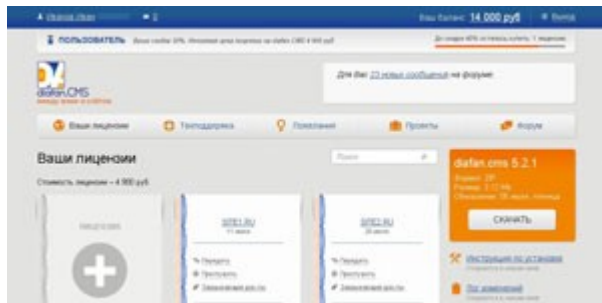
Как зарегистрировать diafan.CMS

- [Что будет, если установить diafan.CMS на домен, на который нет лицензии?](#)

После установки на ваш сайт, diafan.CMS будет работать в пробном режиме 20 дней, не требуя лицензию. Когда данный период закончится, выйдет соответствующее уведомление. Далее необходимо приобрести лицензию на diafan.CMS и вписать в нее домен вашего сайта. Это делается в личном кабинете user.diafan.ru.

[Зарегистрироваться в личном кабинете.](#)

Зарегистрировавшись один раз, можно устанавливать diafan.CMS на множество доменов, просто приобретая лицензии и добавляя имена доменов в личном кабинете.

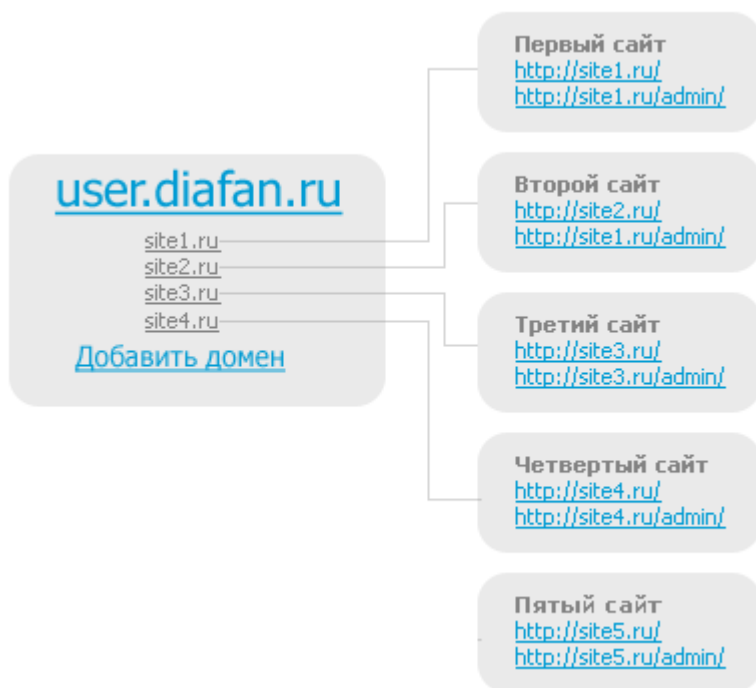


Лицензия для домена действует также и для всех поддоменов, то есть если вы купили лицензию для адреса site.ru, вы сможете использовать diafan.CMS и на сайте с адресом shop.site.ru.

Для тестирования и разработки на локальном веб-сервере лицензия не требуется — создайте для этого локальный адрес без точки — localhost, diafan, mysite и т.п. Автоматическое обновление системы для локальных сайтов невозможно.

Что будет, если установить diafan.CMS на домен, на который нет лицензии?

Схематично связь между личным кабинетом и сайтами можно изобразить так:



На сайты 1, 2, 3 и 4 лицензия есть, поэтому административная часть [site2.ru/admin/](#) работать будет, а вот административная часть сайта 5 без лицензии работать не будет. Т.е. сайтом без лицензии (с незарегистрированным доменом) управлять через [site5.ru/admin/](#) нельзя. При попытке перейти по адресу [site5.ru/admin/](#) будет выведено соответствующее сообщение. Но все пять сайтов по своим основным адресам работать будут. Т.е. сайт [site5.ru](#) открываться будет и пользователи ничего лишнего видеть не будут.

Файловая структура diafan.CMS

- [Что есть что?](#)

После установки diafan.CMS на хостинге вебмастеру открывается следующая структура файлов:

adm
cache
css
img
includes
installation
js
modules
plugins
themes
tmp
userfiles
.htaccess
index.html
index.php
config.php
update.php
update.txt
readme.txt
favicon.ico

Что есть что?

adm – папка панели администрирования. На сайте может иметь любой адрес, по умолчанию *http://site.ru/admin/*, но в процессе инсталляции diafan.CMS этот адрес можно назначить любой, например, *http://site.ru/papka-admina/*. В папке хранится ядро системы управления сайтом, оформление, картинки панели управления, визуальный редактор и вспомогательные функции для управления сайтом. Как правило, менять и править вебмастеру там нечего. Кроме файла *brand.php*, где партнеры-вебстудии могут указать свой логотип, если необходимо клиенту выдать diafan.CMS за свою разработку.

cache – служебная папка CMS, где хранятся файлы дополнительного кеширования, сокращающие SQL-запросы к БД. Файлы создаются автоматически, удаляются так же, поэтому туда можно также особо не заглядывать. Чтобы diafan.CMS успешно работала с файлами, необходимо устанавливать на папку права **777**.

css – папка с CSS-стилями оформления стандартного шаблона дизайна, поставляемого с diafan.CMS, CSS-стили для jQuery-плагинов.

img – папка с картинками оформления стандартного шаблона дизайна, поставляемого с diafan.CMS, изображения для jQuery-плагинов.

includes – здесь хранятся общие функции и вспомогательные алгоритмы для diafan.CMS. Особое внимание стоит обратить лишь на файл *includes/function.php*, в котором собраны

основные функции шаблонных тегов, *show_body*, *show_title* и прочие.

installation – папка инсталляции diafan.CMS.

js – папка с дополнительными JavaScript-скриптами, используемыми на сайте опционально. Это анимация увеличения картинок, всплывающий календарь, загрузчики файлов и т.д.

modules – здесь находятся все модули diafan.CMS. Например, в папке *modules/news* – все файлы модуля «Новости», включая инсталляцию, алгоритм модуля, его оформление и административную часть.

plugins – здесь хранятся [плагины](#) на PHP, используемые в diafan.CMS.

themes – папка шаблонов дизайна сайта. [Основной шаблон](#) дизайна сайта – файл *site.php*. О [шаблонах дизайна](#) читайте далее.

tmp – папка для временных файлов.

userfiles – здесь хранятся все файлы, загружаемые пользователями сайта через административный интерфейс. Картинки из фотогалереи, фотографии к новостям, статьям, загружаемые файлы и изображения. Сразу после установки diafan.CMS папка пустая. Необходимо устанавливать на эту папку и все вложенные в нее элементы права **777**.

.htaccess – управляющий файл diafan.CMS. Достаточно важный файл. Обеспечивает работу псевдоссылок, кодировки, переадресаций и пр.

index.html – файл кэша главной страницы. Раз в сутки заполняется контентом с главной страницы. Если на хостинге происходит сбой и перестает работать БД или интерпретатор, сайт редиректится на этот файл. Таким образом главная страница сайта не бывает «На техобслуживании» из-за хостинга и сбоя, что особенно важно, когда сайт SEO-продвигается. Необходимо устанавливать на этот файл права **777**.

index.php – инициализирующий файл diafan.CMS, стартовый.

config.php – файл конфигурации diafan.CMS. Сохраняет данные из раздела «Параметры сайта» административной части сайта. В случае, если какие-то параметры сбились и вход в административную часть сайта недоступен, можно править параметры вручную в этом файле. Например, изменившиеся параметры подключения к БД. Необходимо устанавливать на этот файл права **777**.

update.php – файл обновления предыдущей версии diafan.CMS до текущей версии.

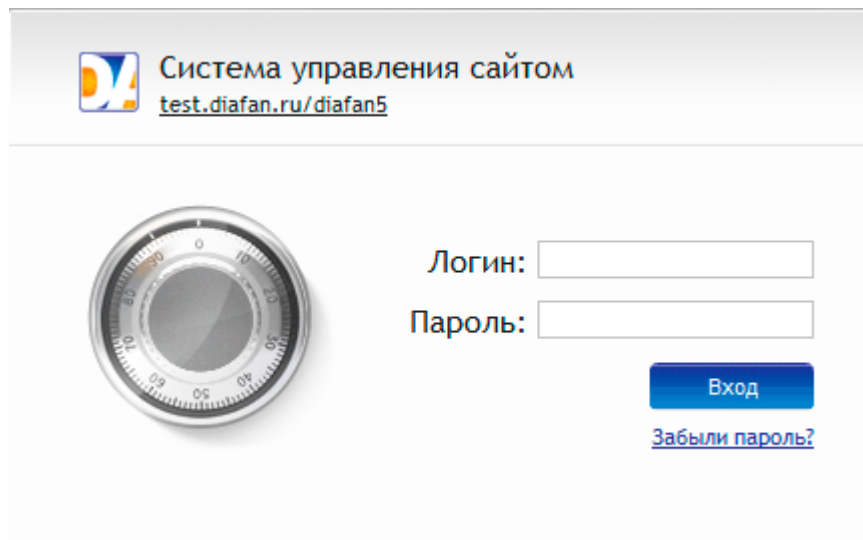
update.txt – руководство по обновлению diafan.CMS с предыдущей версии.

readme.txt – краткое руководство для разработчиков.

favicon.ico – фавикон diafan.CMS.

Вход в панель администрирования diafan.CMS

Набрав в строке адреса браузера (например: Internet Explorer, Firefox, Chrome) адрес вида *http://www.domen.ru/admin/* (вместо *admin* может быть другое имя, уточните у вашего веб-мастера, устанавливавшего diafan.CMS), где *www.domen.ru* – доменное имя вашего сайта, вы попадете на страницу входа в панель администрирования Интернет-сайта:



Система управления сайтом
test.diafan.ru/diafan5

Логин:

Пароль:

[Вход](#)

[Забыли пароль?](#)

Нажав «Вход», вы откроете окно панели администрирования сайта:

Контент

 [Страницы сайта](#)

 [Меню на сайте](#)

 [Новости](#)

 [Фотогалерея](#)

 [Статьи](#)

 [Баннеры](#)

 [Файловый архив](#)

 [Теги](#)

 [Поиск](#)

Интернет магазин

 [Каталог](#)

 [Оплата](#)

 [Заказы](#) ¹

Интерактив

 [Вопрос-Ответ](#) ¹

 [Обратная связь](#) ³

 [Опросы](#)

 [Рейтинг](#)

 [Рассылки](#)

 [Рефералы](#)

 [Комментарии](#)

 [Форум](#)

Управление

 [Пользователи сайта](#)

Страницы сайта

[Главная страница](#)

  [О нас \(м\)](#)

  [Каталог товаров \(м\)](#) 

  [Фотогалерея \(м\)](#) 

  [Вопрос-Ответ \(м\)](#) 

  [Новости \(м\)](#) 

  [Жизнь зоопарка \(м\)](#) 

  [Статьи \(м\)](#) 

  [Обратная связь \(м\)](#) 

  [Форум \(м\)](#) 

  [Контакты \(м\)](#)

   [Наши объекты \(м\)](#)

  [Теги](#) 

  [Поиск по сайту](#) 

☐ [Отметить все](#) [Сделать активными](#)

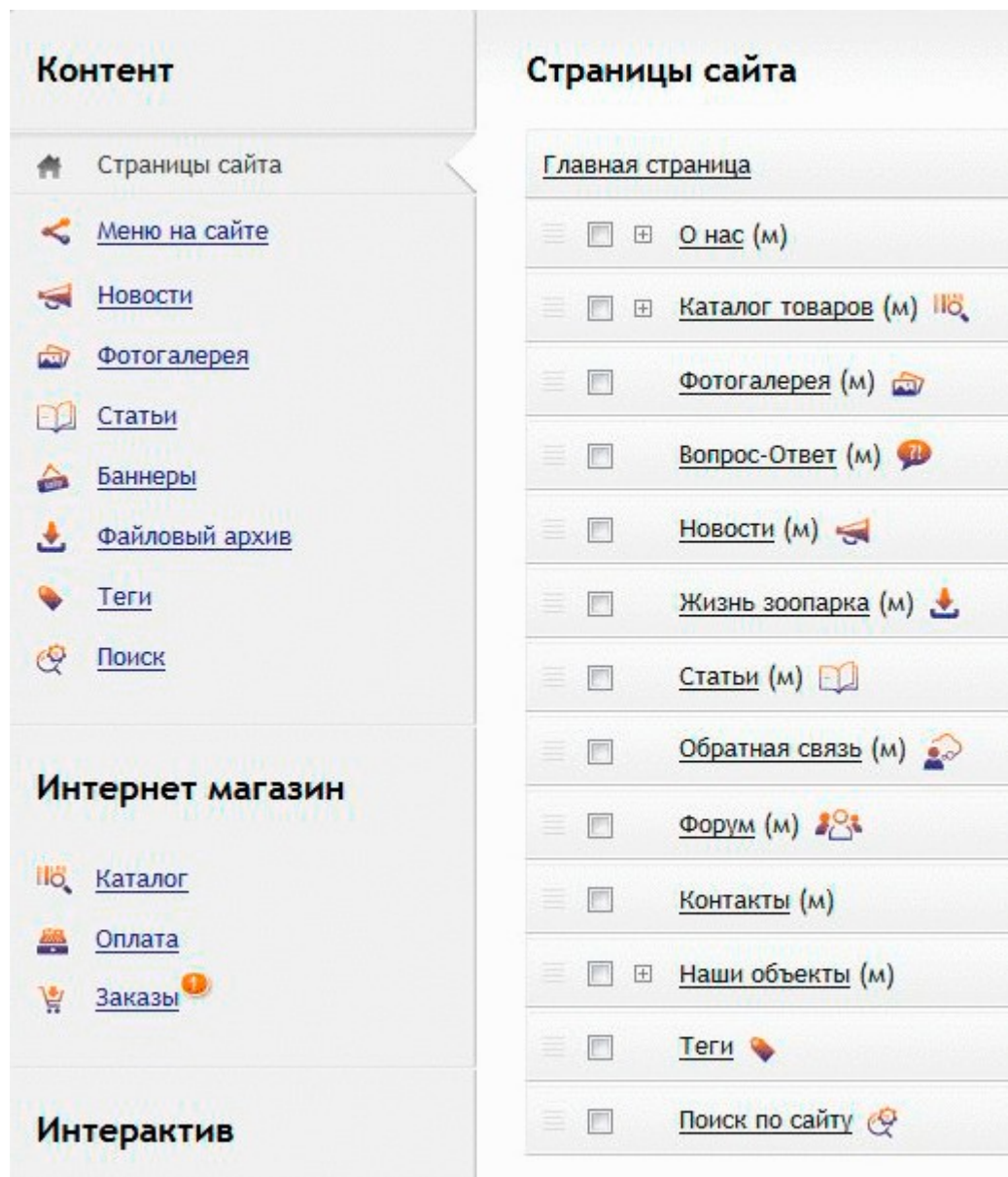
Блоки на сайте ?

[Контакты в шапке](#)

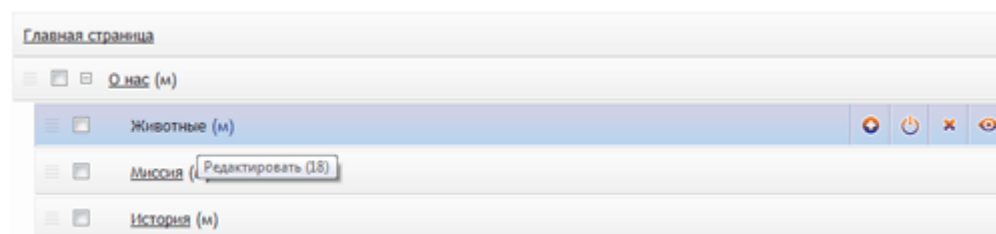
Первое знакомство с панелью управления сайтом

Внимание! [Смотрите видеоурок на тему первого знакомства с diafan.CMS](#)

Панель управления сайтом (для работы подведите курсор мыши к пункту, чтобы увидеть функции):



Страницы сайта: Для добавления раздела или информации в раздел воспользуйтесь пунктом «Страницы сайта».



Наведя курсор мыши на любой из разделов, можно увидеть выплывающее меню, помощь в быстром управлении разделами сайта.

 *Создать подстраницу.* Нажав мышкой на этот знак, можно создать подстраницу в выбранном разделе. Созданный подраздел будет иметь название и такие же возможности редактирования и добавления информации, как и все разделы. Создав подстраницу в разделе сайта, вы увидите слева от названия данного раздела знак +, нажав на который можно просмотреть подразделы, созданные вами (на рисунке показан раскрытый раздел, имеющий два подраздела).

 *Сделать активным / неактивным.* Нажатие на эту кнопку позволяет сделать раздел неактивным (невидимым) на сайте. После нажатия этой кнопки раздел становится серого цвета, а кнопка «Сделать неактивным» сменяется на «Сделать активным». Соответственно, можно вернуть раздел в доступное пользование на сайте, нажав кнопку «Сделать активным».

 *Удалить.* Нажатие на эту кнопку приведет к удалению раздела и всех его подразделов, если таковые имелись. Если выбирать кнопку «Удалить» тот или иной подраздел, удалится только подраздел. Восстановить тот или иной раздел можно будет только создав его заново.

 *Сортировка* пунктов осуществляется перетаскиванием строк мышкой.

 *Посмотреть на сайте.* Нажатие на кнопку «Посмотреть на сайте» позволит посмотреть этот раздел на сайте не выходя из CMS, т.е. в отдельном окне.

Панель редактирования страниц сайта (визуальный редактор)

Для редактирования созданного раздела или подраздела необходимо нажать на название пункта меню в панели администрирования сайта. Откроется следующее окно:

 **Редактировать HTML-код.** Отображает HTML код страницы.

 **Полноэкранный режим.** Раскрывает окно визуального редактора на весь экран.

 **Вырезать.** Чтобы воспользоваться этой кнопкой, необходимо выделить нужную область и нажать на кнопку «Вырезать».

 **Копировать.** Чтобы воспользоваться этой кнопкой, необходимо выделить нужную область и нажать на кнопку «Копировать».

 **Вставить.** Чтобы воспользоваться этой кнопкой, необходимо поставить клавиатурный курсор в ту область текста, куда вы хотите вставить скопированный текст, а затем нажать на кнопку «Вставить». Эта кнопка даёт возможность вставлять информацию в визуальный редактор, сохраняя формат и картинки.

 **Вставить как текст.** Чтобы воспользоваться этой кнопкой, необходимо поставить клавиатурный курсор в ту область текста, куда вы хотите вставить скопированный текст, а затем нажать на кнопку «Вставить».

 **Вставить из Word.** Чтобы воспользоваться этой кнопкой, необходимо поставить клавиатурный курсор в ту область текста, куда вы хотите вставить скопированный текст из MS Word, а затем нажать на кнопку «Вставить из Word». После этого появится дополнительное окошко, в нем вам предлагается вставить текст сочетанием клавиш Ctrl+V, затем нажать «Ок». Для того чтобы сбить настройки форматирования MS Word, необходимо поставить галочку «Убрать определение стилей».

 **Печать.** Кнопка отправления на печать.

 **Отменить (Ctrl+Z).** Позволяет отменить выполненное действие.

 **Вернуть (Ctrl+Y).** Позволяет вернуть отмененное действие.

 **Найти.** Позволяет найти или заменить нужный текст. После нажатия на кнопку «Найти» появится окошко, где вам необходимо ввести искомое слово.

 **Заменить.** Если вам необходимо заменить какое-то слово, выберите в выплывшем окошке вкладку «Заменить».

 **Очистить формат.** Для работы с кнопкой «Убрать форматирование» необходимо выделить нужное слово или воспользоваться кнопкой «Выделить все», если нужно убрать форматирование со всего текста. А затем нажать кнопку «Убрать форматирование».

 **Очистить лишний код.**

 **Правописание.**

 **Цитирование.**

 **Сокращение.**

 **Аббревиация.**

 **Удаление.**

 **Замена.**

В Полужирный (Ctrl+B). Для работы с данными кнопками необходимо выделить нужное слово или сочетание слов и нажать нужную кнопку (сделать текст Жирным, Наклонным, Подчеркнутым, Зачеркнутым).

I Курсив (Ctrl+I). Для работы с данными кнопками необходимо выделить нужное слово или сочетание слов и нажать нужную кнопку (сделать текст Жирным, Наклонным, Подчеркнутым, Зачеркнутым).

U Подчеркнутый (Ctrl+U). Для работы с данными кнопками необходимо выделить нужное слово или сочетание слов и нажать нужную кнопку (сделать текст Жирным, Наклонным, Подчеркнутым, Зачеркнутым).

 **Зачеркнутый.** Для работы с данными кнопками необходимо выделить нужное слово или сочетание слов и нажать нужную кнопку (сделать текст Жирным, Наклонным, Подчеркнутым, Зачеркнутым).

 **Подстрочный.** Чтобы в редактируемом тексте поставить Подстрочный или Надстрочный

индекс, необходимо нажать соответствующую кнопку.

 **Надстрочный.** Чтобы в редактируемом тексте поставить Подстрочный или Надстрочный индекс, необходимо нажать соответствующую кнопку.

 **Цвет текста.** Нажав на данную кнопку, можно изменить цвет шрифта выделенного или набираемого текста.

 **Цвет выделения текста.** Нажав на данную кнопку, можно изменить цвет фона выделенного или набираемого текста.

 **Нумерованный список.** Нажав на данную кнопку, можно создать нумерацию/маркировку выделенного или набираемого текста.

 **Маркированный список.** Нажав на данную кнопку, можно создать нумерацию/маркировку выделенного или набираемого текста.

 **Уменьшить отступ.** Нажав на данные кнопки, можно уменьшить/увеличить отступ текста от края страницы. Также используется для создания вложенных (многоуровневых) списков.

 **Увеличить отступ.** Нажав на данные кнопки, можно уменьшить/увеличить отступ текста от края страницы. Также используется для создания вложенных (многоуровневых) списков.

 **Цитата.** Кнопка позволяет оформить цитату в редактируемом тексте.

 **По левому краю.** Нажатие кнопок позволяет расположить текст по левому краю, центру, правому краю, ширине.

 **По центру.** Нажатие кнопок позволяет расположить текст по левому краю, центру, правому краю, ширине.

 **По правому краю.** Нажатие кнопок позволяет расположить текст по левому краю, центру, правому краю, ширине.

 **По ширине.** Нажатие кнопок позволяет расположить текст по левому краю, центру, правому краю, ширине.

 **Добавить/Изменить ссылку.** Для того чтобы вставить ссылку в редактируемый текст, необходимо выделить нужное слово или сочетание слов и нажать «Добавить/Изменить ссылку» — откроется окно. Заполнив поле URL, вставив нужный адрес и нажав «ОК» вы создадите ссылку. Существует возможность вставки ссылки для скачивания документа. Для добавления документа, доступного для скачивания на сайте, необходимо сделать следующее:

1. Добавить в текст слово или изображение, которое будет являться ссылкой на документ.
2. Выделить данное слово или изображение и нажать кнопку «Добавить/Изменить ссылку».
3. В открывшемся окне нажать кнопку «Обзор», расположенную справа от поля ввода

«Адрес».

4. Выбрать папку на сервере, в которую будет загружен файл. На данном этапе можно создавать вложенные папки, при этом называть их следует используя только латинские буквы и цифры, без пробелов.
5. Нажать «Загрузить файлы», в появившемся окне загрузчика нажать кнопку «Обзор» и выбрать нужный файл на диске вашего компьютера. Имя загружаемого файла должно содержать только латинские буквы и цифры, без пробелов. Если имя файла не удовлетворяет этому требованию, его следует переименовать перед загрузкой.
6. После загрузки файла окно загрузчика можно закрыть.
7. Дважды щелкнуть кнопкой мышки по загруженному файлу.
8. Нажать кнопку «Вставить».

Ссылку на однажды загруженный файл можно вставлять неограниченное количество раз с разных страниц. Для этого на четвертом шаге, после выбора папки на сервере, найдите в ней загруженный ранее файл и перейдите к седьмому шагу.



Удалить ссылку. Для удаления ссылки следует установить на нее клавиатурный курсор и нажать на кнопку «Удалить ссылку».



Добавить/Изменить якорь. Для вставки якоря в редактируемый текст следует воспользоваться данной кнопкой.



Добавить/Изменить изображение (плагин diafan.CMS). Чтобы вставить изображение, необходимо:

1. Поставить клавиатурный курсор туда, где необходимо вставить изображение.
2. На панели редактирования нажать кнопку «Вставить/Редактировать изображение».
3. В появившемся окне можно либо кликнуть на ранее загруженные картинки, либо выбрать картинку с вашего компьютера (для этого необходимо нажать кнопку «Обзор» и в выпавшем проводнике компьютера выбрать нужную картинку, и уже затем кликнуть на неё мышкой).
4. В окне выбрать метод увеличения картинки до большой и нажать «Ок».



Добавить/Изменить клип. Нажав на данную кнопку, можно вставить на страницу вашего сайта файл в формате Flash. В открывшемся окне необходимо заполнить все поля и нажать «Вставить» — Flash появится на странице сайта.



Добавить черту. Нажав на данную кнопку, можно вставить горизонтальную линию на редактируемую страницу сайта.



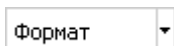
Добавить смайл. Нажав на данную кнопку, можно вставить смайлик на редактируемую страницу сайта.



Добавить символ. Нажав на данную кнопку, можно вставить специальный символ, выбрав его из открывшейся таблицы символов.



Стиль. Нажатие этой кнопки меняет стиль выделенного или набираемого текста.



Формат.



Шрифт. Нажатие этой кнопки позволяет изменить стиль шрифта выделенного или набираемого текста.



Размер. Нажатие этой кнопки позволяет изменить размер шрифта выделенного или набираемого текста.



Добавление/Изменение таблицы. Нажав на данную кнопку, можно вставить таблицу на редактируемую страницу сайта.



Параметры строки.



Параметры ячейки.



Добавить строку сверху.



Добавить строку снизу.



Удалить строку.



Добавить столбец слева.



Добавить столбец справа.



Удалить столбец.



Разбить ячейку.



Объединить ячейки.

После завершения редактирования страницы сайта необходимо нажать одну из кнопок: «Сохранить», «Сохранить и выйти», «Выйти без сохранения»

Если вы хотите сохранить сделанную страницу, следует нажать «Сохранить и выйти». Если вы не хотите сохранить результат работы, следует нажать кнопку «Выйти без сохранения».

Приятной работы в diafan.CMS.

Конфигурация сайта config.php

Конфигурация сайта и основные его параметры сохраняются в файл *config.php*, который лежит в корне сайта. Права файла *config.php* должны быть 777, чтобы diafan.CMS имела возможность его править и сохранять в нем параметры.

Управлять конфигурацией сайта можно с помощью модуля [«Параметры сайта»](#).

Ниже приведена структура файла:

Пример:

```
//папка, в которой лежит сайт. Для корня домена оставьте пустым
define("REVATIVE_PATH", "");

//название сайта, добавляется к тегу title в конце через дефис
define("TIT1", "Бумажный зоопарк");

//параметры подключения к БД
define("DB_URL", "mysql://username:password@localhost/dbname");

//префикс таблиц сайта в БД
define("DB_PREFIX", "diafan_");

//кодировка БД
define("DB_CHARSET", "utf8");

//главный ящик администратора, владельца сайта, используется по умолчанию во всех уведомлениях
define("EMAIL_CONFIG", "mail@site.ru");

//название папки с визуальным редактором
define("USERFILES", "userfiles");

//версия diafan.CMS
define("VERSION_CMS", "5.1");

//дата последнего автообновления в UNIX-формате
define("DATE_UPDATE", "1340192160");

//код обновления
define("DIAFAN_CODE", "");

//ЧПУ папки панели администрирования
define("ADMIN_FOLDER", "admin");

//использовать редактирование сайта с пользовательской части true/false (да/нет)
define("USERADMIN", true);

//включить режим разработки, когда на сайт выводятся все возможные ошибки true/false
```



```
(да/нет)
define("MOD_DEVELOPER", false);

//включить режим технического обслуживания сайта, сайт станет недоступен для
пользователей (шаблон оформления сообщения в /themes/503.php) true/false (да/нет)
define("MOD_DEVELOPER_TECH", false);

//отключить кеширование true/false (да/нет)
define("MOD_DEVELOPER_CACHE", false);

//выводить запросы к БД на сайте true/false (да/нет)
define("MOD_DEVELOPER_PROFILING", false);

//адрес ftp текущего сайта
define("FTP_HOST", "site.ru");

//путь к diafan.CMS, после входа ftp-пользователя, например, www/site.ru/docs/
define("FTP_DIR", "www/site.ru/docs/");

//имя ftp-пользователя
define("FTP_LOGIN", "ftp_username");

//пароль ftp-пользователя
define("FTP_PASSWORD", "ftp_password");

//использовать smtp-авторизацию при отправке почтовых сообщений true/false (да/нет)
define("SMTP_MAIL", true);

//url почтового сервера (например, smtp.mail.ru)
define("SMTP_HOST", "smtp.site.ru");

//логин почты
define("SMTP_LOGIN", "smtp_username");

//пароль к почте
define("SMTP_PASSWORD", "smtp_password");

//порт (по умолчанию 25)
define("SMTP_PORT", "");

//использовать Memcached сервер для кэширования
define("CACHE_MEMCACHED", true);

//хост сервера Memcached
define("CACHE_MEMCACHED_HOST", "localhost");

//порт сервера Memcached
define("CACHE_MEMCACHED_PORT", "11211");

//часовой пояс сайта, в формате http://www.php.net/manual/en/timezones.php
define("TIMEZONE", "Europe/Moscow");
```

//конец строки ЧПУ, по умолчанию "/". Можно ввести ".htm"

```
define("ROUTE_END", "/");
```

//использовать автоматическое формирование ЧПУ для модулей true/false (да/нет)

```
define("ROUTE_AUTO_MODULE", true);
```

//дата последнего экспорта заказов в систему 1С:Предприятие

```
define("LAST_1C_EXPORT", "1.10.2012 13:21");
```

Интеграция дизайна, шаблоны и шаблонные функции

- [Шаблоны](#)
- [Шаблонные функции и теги](#)

Шаблоны

Шаблоны – это оформление сайта.

[Шаблон\(ы\) сайта](#) – это оформление страниц(ы) сайта в пользовательской части. По сути, это и есть традиционный дизайн всего сайта. Всё то, что видят посетители сайта. Шаблоны сайта хранятся в папке themes.

[Шаблоны модулей](#) – это файлы `modules/имя_модуля/views/имя_модуля.view.название_шаблона.php`. Шаблоны модулей отвечают за оформление вывода модульной информации: список новостей, товаров, статей, карточки товаров, голосования и т.д.

Шаблонные функции и теги

Шаблонные функции – это функции для вывода всевозможного контента в шаблонах сайта. Шаблонная функция вызывается [шаблонным тегом](#), который является основным инструментом при [подключении дизайна к diafan.CMS](#). Шаблонные функции делятся на:

- [общие шаблонные функции](#) – это функции из файла `includes/function.php` для общего назначения;
- [шаблонные функции модулей](#) – это функции, хранящиеся в шаблоне модуля `modules/имя_модуля/views/имя_модуля.view.название_шаблона.php` и служат для вывода на сайте модульной информации, новостей, статей, товаров, голосований и т.д.

Как подключить diafan.CMS к дизайну

Внимание! [Смотрите видеоурок на тему подключения дизайна к diafan.CMS](#)

Все очень просто! Подключить дизайн к diafan.CMS – легче легкого! :)

[Ставите CMS](#) к себе на сайт. У вас сразу все будет работать. Но на вашем сайте будет дизайн по умолчанию, как в демо-версии.

Это стандартный шаблон, *themes/site.php*. По сути, это простой, обычный HTML-файл, но с нашими шаблонными тегами. Его можно удалить, а еще лучше переименовать, и затем оттуда брать за основу теги для внедрения своего дизайна.

Пример:

Шаблонный тег в diafan.CMS – это тег, который запускает шаблонную функцию, которая выводит необходимые данные из CMS в HTML-дизайне.

Если загрузить любой HTML-файл в папку *themes* как файл *site.php*, то diafan.CMS содержимое этого файла сразу выведет на сайте как дизайн.

Чтобы в дизайне выводилось меню из CMS, строка навигации, тексты из админки, блоки новостей, теги *title* и т.д., необходимо указать diafan.CMS, что где в HTML-коде выводить. Это как раз делается с помощью [шаблонных функций \(тегов\)](#).

Устроен шаблонный тег так:

`<insert name="..." атрибуты>`

Например, чтобы вывести меню, нужно вставить в файл *site.php* шаблонный тег

`<insert name="show_block" module="menu">`

Для вывода контента сайта служит тег

`<insert name="show_body">`

И так далее.

Рассмотрим подробнее. Допустим, у вас дизайн сайта с сеткой из трех колонок:

Логотип

[Меню 1](#) ★ [Меню 2](#) ★ [Меню 3](#) ★ [Меню 4](#)

Новости

01.01.2009
[Название](#)
Анонс новости 1

02.01.2009
[Название](#)
Анонс новости 2

Опрос

Как вам наш сайт?
Нравится
Не нравится
Все равно
[Ответить](#)

[Главная](#) - [Меню 1](#)

Заголовок

Текст сайта
Добро пожаловать
Рады вас видеть
Наш сайт открылся
Читайте у нас новости и статьи

Статьи

[Название](#)
Анонс статьи 1

[Название](#)
Анонс статьи 2

Облако тегов

[Каталог](#) [Товар](#)
[Скидка](#) [Заказ](#)

HTML-код этого дизайна примерно такой:

Пример:

```
<title>Заголовок сайта</title>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
```

```
<meta name="description" content="Описание">
```

```
<meta name="keywords" content="Ключевики">
```

```
</head>
```

```
<body>
```

```
<table>
```

```
<tr>
```

```
<td>
```

```
Логотип
```

```
</td>
```

```
<td>
```

```
<a href="#">Меню 1</a>
```

```
<a href="#">Меню 2</a>
```

```
<a href="#">Меню 3</a>
```

```
<a href="#">Меню 4</a>
```

```
</td>
```

```
</tr>
```

```
</table>
```

```
<table>
```

```
<tr>
```

```
<td>
```

```
Новости<br>
```

```
Опрос
```

```
</td>
```

```
<td>
```

```
Строка навигации<br>
```

```
Текст сайта
```

```
</td>
```

```

        <td>
        Статьи<br>
        Облако тегов
        </td>
    </tr>
</table>
</body>
</html>

```

Первым делом сохраните этот HTML файл как *themes/site.php*. Вы сразу увидите, что он отобразится на сайте! Но ссылки останутся #, тег *title* так и будет «Заголовок сайта» и т.д. Чтобы все заработало, расставьте шаблонные теги так:

Пример:

```

<html>
<head>

    <title><insert name="show_title"></title>

    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
    <meta name="description" content="<insert name="show_description">">
    <meta name="keywords" content="<insert name="show_keywords">">
</head>
<body>
    <table>
    <tr>
        <td>
            Логотип
        </td>
        <td>
            <insert name="show_block" module="menu" separator_1="[img src=`razdelitel.gif`]">
        </td>
    </tr>
    </table>
    <table>
    <tr>
        <td>
            <insert name="show_block" module="news" count="3">
            <insert name="show_block" module="votes">
        </td>
        <td>
            <insert name="show_path">
            <insert name="show_body">
        </td>
        <td>
            <insert name="show_block" module="clauses">
            <insert name="show_block" module="tags">
        </td>
    </tr>
    </table>
</body>

```

</html>

Где:

- `<insert name="show_block" module="menu" separator_1="[img src=razdelitel.gif ]">` – шаблонный тег вывода пунктов меню. Атрибут `separator_1` в данном случае нужен, чтобы функция выводила `` между пунктами меню;
- `<insert name="show_block" module="news" count="3">` – шаблонный тег вывода блока анонсов новостей, трех последних;
- `<insert name="show_block" module="votes">` – шаблонный тег вывода опросников;
- `<insert name="show_path">` – шаблонный тег вывода строки навигации «Хлебные крошки»;
- `<insert name="show_body">` – шаблонный тег вывода основного контента сайта;
- `<insert name="show_block" module="clauses">` – шаблонный тег вывода анонсов статей;
- `<insert name="show_block" module="tags">` – шаблонный тег вывода облака тегов.

Полное описание всех шаблонных тегов и их параметров смотрите [здесь](#).

В diafan.CMS используются JQuery, JavaScript, стили оформления форм и текстов, поэтому, если эти функции, стили и скрипты нужны, или вы хотите оставить функции по умолчанию, необходимо оставить тег `<insert name="show_js">` между тегами `<head></head>` из исходного `themes/site.php`:

Пример:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>

<title><insert name="show_title"></title>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<meta name="description" content="<insert name="show_description">">
<meta name="keywords" content="<insert name="show_keywords">">

<meta content="<insert value="Russian">" name="language">
<meta content="Diafan http://www.diafan.ru" name="author">
<meta content="All" name="robots">

<insert name="show_js">

<link rel="alternate" type="application/rss+xml" title="RSS" href="<insert
name="path_url">news/rss/">

<link href="<insert name="path">css/style.css" rel="stylesheet" type="text/css">

</head>
<body <insert name="show_protect">>

</body>
```

</html>

А свой HTML-код вставляйте между тегами <body></body>.

Шаблонный тег <insert name="show_js"> выводит список подключаемых ява-скриптов. Изменить этот список можно в файле *includes/function.php*.

Если для какой-то отдельной страницы на сайте необходимо иметь другой дизайн, сохраните его в папке *themes*. Затем при редактировании страницы в панели управления в «Дополнительно» укажите его в выпадающем списке «Шаблон дизайна».

Шаблон сайта

- [Мультидизайн](#)

Шаблон сайта – это оформление страниц(ы) в пользовательской части сайта, дизайн сайта.

Основной шаблон сайта – *themes/site.php*. По умолчанию все страницы сайта используют файл шаблона именно с этим именем, если какой-то странице не назначить принудительно шаблон с другим именем (см.ниже). Шаблон представляет собой файл с обычным [HTML](#)-кодом со вставками [шаблонных тегов](#) diafan.CMS.

Внимание! При изменении дизайна сайта проще всего взять исходный файл, входящий в пакет diafan.CMS по умолчанию *themes/site.php* и заменить HTML-код, находящийся между тегами: `<body></body>`

Мультидизайн

diafan.CMS поддерживает мультидизайн сайта. Это означает, что разные страницы пользовательской части сайта могут иметь разный внешний вид.

Чтобы какой-либо странице сайта установить **другой дизайн**, отличный от *themes/site.php*, необходимо:

1. В папку *themes* загрузить шаблон(ы) с расширением .php (например, *site_shop.php* или *list_news.php*). Имя шаблона не может содержать никаких символов, кроме букв латинского алфавита, точки, тире и нижнего подчеркивания.
2. В административной части сайта (модуль «Страницы сайта») открыть страницу, для которой необходимо подключить другой дизайн и раскрыть вкладку «Дополнительно». Diafan.CMS просканирует папку *themes*, найдет там все ваши загруженные файлы, и в выпадающем списке свойства «Дизайн страницы» останется только выбрать загруженный файл шаблона.

«Дизайн страницы» наследуется страницами нижнего уровня по умолчанию.

Шаблоны модулей

Шаблон модуля – это набор файлов в папке *modules/имя_модуля/views*. Шаблоны модулей отвечают за локальное оформление сайта в основной контентной зоне сайта. Как правило, это список новостей, товаров, файлов, фотографий, а также отдельные страницы статей, карточки товаров, блоки голосований, формы отзывов и т.д.

Внимание! Если необходимо изменить оформление даты новостей на сайте, то нужно идти в шаблон модуля новости: *modules/news/views/news.view.list.php*

Обычно состав файлов следующий:

- *модуль.view.form.php* – выводит в пользовательской части форму (если модуль интерактивный). Например, в модуле «Вопрос-ответ» форму отправки вопроса;
- *модуль.view.list.php* – выводит в пользовательской части список элементов модуля. Например, в модуле «Новости» – список новостей;
- *модуль.view.first_page.php* – выводит главную страницу модуля;
- *модуль.view.id.php* – выводит в пользовательской части отдельный элемент. Например, отдельную новость в модуле «Новости»; или отдельный товар в модуле «Магазин»;
- *модуль.view.show_block.php* – выводит шаблонную функцию модуля. Например, для модуля «Новости» – вывод блока новостей для шаблонного тега [show_block](#).

В шаблоне могут быть другие файлы, отражающие индивидуальные особенности модуля.

Подключение шаблона происходит следующим образом:

Пример:

```
$this->diafan->_tpl->get('название_шаблона', 'название_модуля', $result);
```

где *\$result* – передаваемые в шаблон данные в виде массива.

В шаблоне для вызова другого шаблона можно воспользоваться сокращенной формой:

Пример:

```
$this->get('название_шаблона', 'название_модуля', $result);
```

В шаблоне модуля можно вызвать шаблонный тег. Для этого его нужно обработать функцией *htmleditor*.

Пример:

```
$this->htmleditor('<insert name="show_block" module="site" id="13">');
```

Также, все данные, вводимые с помощью визуального редактора, нужно обрабатывать с помощью этой функции.

Пример:

```
$this->htmleditor($row["text"]);
```

Шаблонные функции и теги

- [Шаблонные теги](#)
- [Общие шаблонные функции](#)

Шаблонные функции – это функции для вывода всевозможного контента в пользовательской части сайта. Шаблонная функция неразрывна с шаблонным тегом, который является инструментом для вызова функции в шаблонах сайта при [подключении дизайна к diafan.CMS](#).

Шаблонные теги

Шаблонные теги служат для запуска шаблонных функций, которые выводят различный контент в шаблонах сайта. Шаблонный тег представляет собой тег вида:

```
<insert [name=""] [module=""] [value=""] [attribute_1=""] [attribute_2=""]>
```

Тег включает ряд необязательных атрибутов, делящихся на определенные и неопределенные атрибуты.

Определенные атрибуты:

- **name** – задает вызываемую функцию или выводимую переменную;
- **value** – значение выводится на странице, для мультязычного сайта можно задать перевод значения;
- **module** – задает название модуля, к которому относится вызываемая функция.

Неопределенные атрибуты – это все остальные атрибуты. Они передаются функции в качестве массива в формате:

```
array(название => значение, ...);
```

Значения атрибутов должны быть заключены в кавычки и не должны содержать символы >, <, ", '. Эти символы должны быть заменены HTML-мнемониками или специальными символами:

- < надо заменить на `<`; или [
- > надо заменить на `>`; или]
- & надо заменить на `&`;
- " надо заменить на `"`; или `

Шаблонный тег может быть оформлен и выводить:

1. Значение объявленных переменных:

- **path** – адрес сайта;

Пример:

```
<insert name="path">
```

- **path_url** – адрес сайта, с учетом языка сайта;

Пример:

```
<insert name="path_url">
```

- **language** – идентификатор языка сайта;

Пример:

```
<insert name="language">
```

- **title** – название сайта на текущем языке;

Пример:

```
<insert name="title">
```

- **module** – название модуля, прикрепленного к текущей странице;

Пример:

```
<insert name="module">
```

2. Переводимый текст для мультязычных сайтов (**value="значение"**);

Пример:

```
<insert value="Русский">
```

выведет слово «Русский» в основной языковой версии сайта. Для других языковых версий перевод можно задать в административной части сайта в разделе [языки сайта](#) или перевести слово в пользовательской части сайта при помощи [панели быстрого редактирования](#).

3. Контент, генерируемый [общими шаблонными функциями](#);

Пример:

```
<insert name="show_year" year="2003">
```

выведет на сайте 2003-2013

4. Контент, генерируемый [шаблонными функциями модулей](#).

Пример:

```
<insert name="show_block" module="menu" tag_start_1="[span class=`menutd`]"  
tag_end_1="[/span][br]" tag_start_2="[span class=`menutd_level`]" tag_end_2="[/span]  
[br]">
```

выведет меню.

Общие шаблонные функции

Это функции общего назначения из файла *includes/function.php*:

- [show_body](#) – выводит основной контент страницы: заголовок (если не запрещен его вывод в настройке страницы «Не показывать заголовок»), текст страницы и прикрепленный модуль. Заменяет три тега: *show_h1*, *show_text*, *show_module*;
- [show_description](#) – выводит мета-тег description страницы;
- [show_h1](#) – выводит заголовок страницы, если не запрещен его вывод в настройке страницы «Не показывать заголовок»;
- [show_href](#) – выводит ссылку на страницу сайта. Если текущая страница соответствует адресу, на который ведет ссылка, то ссылка становится неактивной. Шаблонная функция аналогична стандартной HTML-конструкции `<a href=""></a>`, но ликвидирует ссылки, которые никуда не ведут;
- [show_include](#) – подключает файл-блок шаблона;
- [show_js](#) – подключает JS файлы, обеспечивающие работу diafan.CMS;
- [show_keywords](#) – выводит ключевые слова страницы. Используется для мета-тега keywords;
- [show_module](#) – выводит контент модуля;
- [show_path](#) – выводит навигацию по сайту «Хлебные крошки»;
- [show_protect](#) – выводит JS код, запрещающий копирование контента на сайте;
- [show_social_links](#) – выводит ссылки на социальные сети;
- [show_text](#) – выводит текст страницы;
- [show_title](#) – выводит заголовок. Используется между тегами `<title></title>` в шапке сайта;
- [show_year](#) – выводит период функционирования сайта в годах.

Шаблонные функции модулей

Шаблонные функции модулей – это функции для вывода в пользовательской части сайта всевозможного контента из модулей. Шаблонная функция модуля также неразрывна с шаблонным тегом, который является инструментом для вызова функции в шаблонах сайта при [подключении дизайна к diafan.CMS](#).

Описание шаблонных функций модулей дано в описании каждого модуля.

[On-line консультант](#)

[Баннеры](#)

[Вопрос-ответ](#)

[Магазин](#)

[Меню на сайте](#)

[Новости](#)

[Обратная связь](#)

[Объявления](#)

[Опросы](#)

[Ошибка на сайте](#)

[Поиск по сайту](#)

[Пользователи](#)

[Рассылки](#)

[Статьи](#)

[Страницы сайта](#)

[Теги](#)

[Файловый архив](#)

[Форум](#)

[Фотогалерея](#)

[Языки сайта](#)

А также есть [общий справочник](#), где собраны все шаблонные теги.

Системные модули

Системные модули – это модули, входящие в состав ядра diafan.CMS, их нельзя удалить, и устанавливаются они в любом случае.

Страницы сайта

Основной модуль системы. Позволяет создавать структуру неограниченной вложенности, наполнять сайт контентом с помощью визуального редактора. Автоматически генерирует заголовок страницы, поддерживает ЧПУ. Модуль дает возможность управлять SEO-тегами, корректно отдает заголовки, типа Last-Modify, и многое другое.

Меню на сайте

Формирование меню любой сложности и в любом количестве.

Пользователи сайта

Редактирование пользователей сайта и их прав доступа.

Обновление CMS

Обновление сайта в пределах минорной версии diafan.CMS, список закрытых для обновления файлов.

Модули и БД

Установка и удаление модулей, восстановление базы данных, экспорт и импорт базы данных.

Параметры сайта

Редактирование общих настроек сайта.

Языки сайта

Управление языковыми версиями сайта, перевод интерфейса.

Корзина

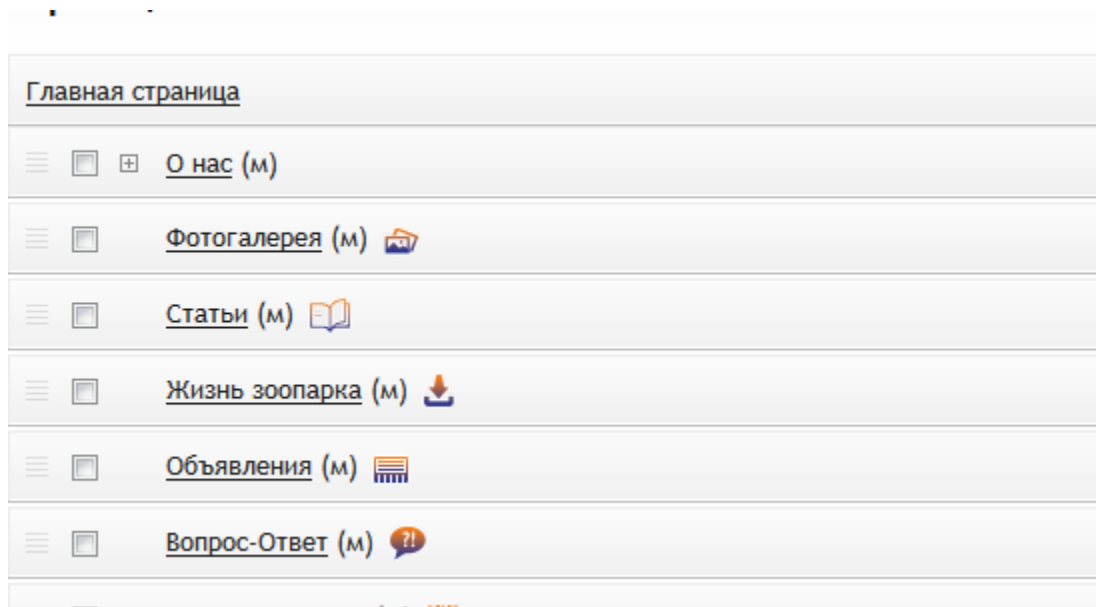
Управление удаленными элементами.

Страницы сайта

- [Блоки на сайте](#)
- [Настройки](#)
- [Шаблонные теги](#)
- [Файлы](#)

Модуль «Страницы сайта» – основной механизм управления сайтом.

Модуль относится к части CMS «Контент» и является обязательным для установки.



Если перед названием страницы имеется знак +, то этой странице принадлежат вложенные страницы нижнего уровня.

При наведении на раздел появляются 4 кнопки

Добавить страницу нижнего уровня можно, нажав на знак  рядом с названием текущей страницы в списке всех страниц.

Кнопка  служит для удаления текущей страницы. Если странице принадлежат страницы нижнего уровня, то при удалении они будут также удалены.

Кнопка  служит для быстрой публикации страницы (или, наоборот, блокирования).

Кнопка  открывает страницу в пользовательской части сайта.

Сортировать страницы можно простым перетаскиванием мышкой, удерживая строку слева за ребрышки .

Если страница выведена в меню, то после нее будет выведена метка «(м)».

Если страница помечена опцией «Блок на сайте», то она будет выведена внизу списка под заголовком «Блоки на сайте».

загрузки изображения будут обработаны автоматически, согласно настройкам модуля. Параметр выводится, если в настройках модуля отмечена опция «Использовать изображения».

- **Комментарии** – Комментарии, которые оставили пользователи к текущей странице. Параметр выводится, если в настройках модуля включен параметр «Показывать комментарии к новостям».
- **Контент страницы** – Основное содержимое страницы. Если отметить «Применить типограф», контент будет отформатирован согласно правилам экранной типографики с помощью [веб-сервиса «Типограф»](#). Параметр имеет разные значения для разных языковых версий сайта.
- **Индексирование для поиска** – Страница автоматически индексируется для модуля «Поиск по сайту» при внесении изменений.
- **Индексирование для карты сайта** – Страница автоматически индексируется для карты сайта *sitemap.xml*.

Дополнительные:

- **Номер** – Номер страницы в БД (веб-мастеру и программисту).
- **Редактор** – Изменяется после первого сохранения. Показывает, кто из администраторов сайта первый правил текущую страницу.
- **Время последнего изменения** – Изменяется после сохранения элемента. Отдается в заголовке *Last Modify*.
- **Заголовок окна в браузере, тег Title** – Если не заполнен, тег *Title* будет автоматически сформирован как «Название страницы – Название сайта», либо согласно шаблону автоформирования из настроек модуля (SEO-специалисту). Параметр имеет разные значения для разных языковых версий сайта.
- **Ключевые слова, тег Keywords** – Если не заполнен, тег *Keywords* будет автоматически сформирован согласно шаблону автоформирования из настроек модуля (SEO-специалисту). Параметр имеет разные значения для разных языковых версий сайта.
- **Описание страницы, тег Description** – Если не заполнен, тег *Description* будет автоматически сформирован согласно шаблону автоформирования из настроек модуля (SEO-специалисту). Параметр имеет разные значения для разных языковых версий сайта.
- **Не копировать автоматически название в заголовок H1** – Если отмечено, заголовок *H1* перед текстом страницы автоматически выводиться не будет. Тогда его можно вписать в визуальный редактор в свободном виде (SEO-специалисту).
- **Псевдоссылка** – ЧПУ, т.е. адрес страницы вида: <http://site.ru/pseudossylka/>. Смотрите параметры сайта (SEO-специалисту).
- **Не индексировать** – Запрет индексации текущей страницы, если отметить, у страницы выведется тег: `<meta name="robots" content="noindex">` (SEO-специалисту).
- **Changefreq** – Вероятная частота изменения этой страницы. Это значение используется для генерирования файла *sitemap.xml*. Подробнее читайте в описании [XML-формата файла Sitemap](#) (SEO-специалисту).
- **Priority** – Приоритетность URL относительно других URL на Вашем сайте. Это значение используется для генерирования файла *sitemap.xml*. Подробнее читайте в описании [XML-формата файла Sitemap](#) (SEO-специалисту).
- **Период показа** – Если выставить, текущая страница будет опубликована на сайте в указанный период. В иное время пользователи сайта страницу не будут видеть, получая ошибку 404 «Страница не найдена» (администратору сайта).
- **Доступ** – Если отметить опцию «Доступ только», страницу увидят только

авторизованные на сайте пользователи, отмеченных типов. Не авторизованные, в том числе поисковые роботы, увидят «404 Страница не найдена» (администратору сайта).

- **Сортировка: установить перед** – Изменить положение текущей страницы среди других страниц (администратору сайта).
- **Не показывать на карте сайта** – Если отмечено, страница не будет показываться на карте сайта (администратору сайта).
- **Не показывать в результатах поиска по сайту** – Страница не участвует в поисковой выдаче внутреннего поиска по сайту.
- **Прикрепить модуль** – Прикрепление модуля к текущей странице. Содержимое модуля выведется после контента страницы (администратору сайта).

Пример:

Если у нас обычная страница <http://site.ru/222/> с текстом «привет» и мы прикрепим к ней модуль новости, то под текстом страницы «привет» ниже начнет выводиться список новостей. На каждую новость можно нажать и перейти к ней, ее адрес сформируется по правилу <http://site.ru/222/novost1/> или <http://site.ru/222/kategoria/novost1/>. То есть модуль станет работать на странице <http://site.ru/222/> и сформирует свои страницы вглубь.

- **JavaScript** – Поле для ввода JavaScript на текущей странице. Например, для кода Яндекс.Карт. ВНИМАНИЕ, не вставляйте JS-код в визуальный редактор, он может его обрезать. Вставляйте его в это поле. (Веб-мастеру и программисту).
- **Дизайн страницы** – Возможность подключить для страницы шаблон сайта отличный от основного (*themes/site.php*). Все шаблоны для сайта должны храниться в папке *themes* с расширением *.php* (например, *themes/dizain_so_slajdom.php*). Подробнее в [разделе «Шаблоны сайта»](#). (веб-мастеру и программисту, не меняйте этот параметр, если не уверены в результате!).

Блоки на сайте

Блок на сайте – это контент, который может быть выведен шаблонным тегом [show_block](#) в любом месте шаблона.

Основные:

- **Название блока** – Название блока, выводится перед содержимым блока, если не отмечена галка «Не выводить название блока». Параметр имеет разные значения для разных языковых версий сайта.
- **Опубликовать на сайте** – Отображение блока на сайте. Если не отмечена, блок на сайте не будет показываться. Параметр имеет разные значения для разных языковых версий сайта.
- **Отображать на страницах** – Выбор отдельных страниц, где будет показываться блок. Удерживайте CTRL, чтобы выбрать несколько страниц.
- **Не выводить название блока** – Если отмечено, заголовок перед содержимым блока автоматически выводиться не будет.
- **Контент блока** – Основное содержимое блока. Если отметить «Применить типограф», контент будет отформатирован согласно правилам экранной типографики с помощью [веб-сервиса «Типограф»](#). Параметр имеет разные значения для разных языковых версий сайта.

Дополнительные:

- **Номер** – Номер элемента в БД (веб-мастеру и программисту).
- **Редактор** – Изменяется после первого сохранения. Показывает, кто из администраторов сайта первый правил текущий блок.
- **Время последнего изменения** – Изменяется после сохранения элемента. Отдается в заголовке *Last Modify*.
- **Доступ** – Если отметить опцию «Доступ только», блок увидят только авторизованные на сайте пользователи, отмеченных типов (администратору сайта).
- **Период показа** – Если выставить, текущий блок будет опубликован на сайте в указанный период. В иное время пользователи сайта блок не будут видеть (администратору сайта).
- **Сортировка: установить перед** – Изменить положение текущего блока среди других блоков. Используется для удобства администрирования блоков (администратору сайта).

Настройки

Настройки

Использовать изображения ☒ ?

Генерировать [размеры изображений](#) Большое изображение (полная версия) ▼ Маленькое изображение (превью) ▼

Использовать анимацию при увеличении изображений ☒

Максимальный размер загружаемых файлов 20М

Применить настройки ко всем ранее загруженным изображениям

- **Защита от копирования** – Если отмечена, на сайте будет отключена возможность копирования текста в буфер. См. шаблонную функцию `show_protect`.
- **Прикрепить комментарии к страницам сайта** – Если отмечена, пользователи сайта будут иметь возможность оставлять комментарии к каждой странице сайта. Комментарии для страницы сайта выводятся с помощью шаблонной функции `show_comments`. Комментарии выводятся только на страницах, к которым не подключен иной модуль со своими комментариями. Параметр не будет включен, если модуль «Комментарии» не установлен. Подробности см. в разделе [модуль Комментарии](#).
- **Подключить теги** – Если отмечена, к страницам сайта можно будет прикреплять теги. Теги для страницы сайта выводятся с помощью шаблонной функции `show_tags`. Параметр не будет включен, если модуль «Теги» не установлен. Подробности см. в разделе [модуль «Теги»](#).
- **Использовать изображения** – Если отмечена, к страницам сайта можно будет прикреплять дополнительные изображения. Изображения для страницы сайта выводятся с помощью шаблонной функции `show_images`.

- **Генерировать размеры изображений** – Размеры изображений, заданные в модуле «Изображения» и тег латинскими буквами для подключения изображения на сайте. Параметр выводится, если отмечена опция «Использовать изображения».
- **Максимальный размер загружаемых файлов** – Параметр показывает максимально допустимый размер загружаемых файлов, установленный в настройках хостинга. Параметр выводится, если отмечена опция «Использовать изображения».
- **Подключить перелинковку** – Отображение перелинковки в модуле. Подробности см. в разделе [модуль «Перелинковка»](#).
- **Шаблон для автоматического генерирования Title** – Если шаблон задан, то заголовок автоматически генерируется по шаблону. В шаблон можно добавить:
 - %title – заданный заголовок,
 - %name – название,
 - %parent – название страницы верхнего уровня (SEO-специалисту).
 Параметр имеет разные значения для разных языковых версий сайта.
- **Шаблон для автоматического генерирования Keywords** – Если шаблон задан, то поле *Keywords* автоматически генерируется по шаблону. В шаблон можно добавить:
 - %keywords – заданные ключевые слова,
 - %name – название,
 - %parent – название страницы верхнего уровня (SEO-специалисту).
 Параметр имеет разные значения для разных языковых версий сайта.
- **Шаблон для автоматического генерирования Description** – Если шаблон задан, то поле *Description* автоматически генерируется по шаблону. В шаблон можно добавить:
 - %descr – заданное описание,
 - %name – название,
 - %parent – название страницы верхнего уровня (SEO-специалисту).
 Параметр имеет разные значения для разных языковых версий сайта.

Шаблонные теги

Для работы с модулем «Страницы сайта» служат следующие [шаблонные теги](#):

show_block – выводит содержимое страницы сайта, номер которой передан в виде атрибута id. Страница должна быть помечена опцией «Блок на сайте» и быть активной.

Атрибуты:

— **id** – идентификатор блока;

— **template** – шаблон тега (файл *modules/site/views/site.view.show_block_template.php*; по умолчанию шаблон *modules/site/views/site.view.show_block.php*).

Пример:

```
<insert name="show_block" module="site" id="7">
```

выведет в ячейке таблицы содержимое страницы с id=7, редактируемой в модуле «Страницы сайта»

show_comments – выводит комментарии, прикрепленные к странице сайта, если в конфигурации модуля «Страницы сайты» подключены комментарии.

Атрибуты:

— **template** – шаблон тега (файл *modules/site/views/site.view.show_comments_***template***.php*; по умолчанию шаблон *modules/site/views/site.view.show_comments.php*).

Пример:

```
<insert name="show_comments" module="site">
```

выведет форму для отправки комментариев и добавленные комментарии посетителей

show_images – выводит изображения, прикрепленные к странице сайта, если в конфигурации модуля «Страницы сайта» включен параметр «Использовать изображения».

Атрибуты:

— **template** – шаблон тега (файл *modules/site/views/site.view.show_images_***template***.php*; по умолчанию шаблон *modules/site/views/site.view.show_images.php*).

Пример:

```
<insert name="show_images" module="site">
```

выведет в ячейке таблицы группу фотографий, прикрепленных к текущей странице

show_links – выводит ссылки на страницы нижнего уровня, принадлежащие текущей странице.

Атрибуты:

— **template** – шаблон тега (файл *modules/site/views/site.view.show_links_***template***.php*; по умолчанию шаблон *modules/site/views/site.view.show_links.php*).

Пример:

```
<insert name="show_links" module="site">
```

show_previous_next – выводит ссылки на предыдущую и последующую страницы.

Атрибуты:

— **template** – шаблон тега (файл *modules/site/views/site.view.show_previous_next_***template***.php*; по умолчанию шаблон *modules/site/views/site.view.show_previous_next.php*).

Пример:

```
<insert name="show_previous_next" module="site">
```


show_tags – выводит теги (слова-якори), прикрепленные к странице сайта, если в конфигурации модуля «Страницы сайты» подключены теги.

Атрибуты:

— **template** – шаблон тега (файл *modules/site/views/site.view.show_tags_template.php*; по умолчанию шаблон *modules/site/views/site.view.show_tags.php*).

Пример:

```
<insert name="show_tags" module="site">
```

выведет список ссылок на теги, прикрепленные к текущей странице

Файлы

1. *modules/site/admin/site.admin.php* – редактирование страниц сайта;
2. *modules/site/admin/site.admin.blocks.php* – редактирование блоков на сайте;
3. *modules/site/admin/site.admin.config.php* – настройки модуля;
4. *modules/site/site.php* – контроллер;
5. *modules/site/site.model.php* – модель;
6. *modules/site/site.search.php* – настройки для поисковой индексации для модуля «Поиск»;
7. *modules/site/views/site.view.show_block.php* – шаблон блока на сайте;
8. *modules/site/views/site.view.show_comments.php* – шаблон комментариев к странице сайта;
9. *modules/site/views/site.view.show_images.php* – шаблон изображений к странице сайта;
10. *modules/site/views/site.view.show_links.php* – шаблон вложенных страниц сайта;
11. *modules/site/views/site.view.show_previous_next.php* – шаблон ссылок на предыдущую и следующую страницы сайта;
12. *modules/site/views/site.view.show_tags.php* – шаблон тегов к странице сайта.

Меню на сайте

- [Пункты меню](#)
- [Меню](#)
- [Настройки](#)
- [Шаблонные теги](#)
- [Файлы](#)

Модуль «Меню на сайте» формирует меню любой сложности и в любом количестве.

Модуль состоит из 2х частей: пункты меню и меню.

Пункты меню

Пункты меню – ссылки, формирующие меню. То есть, по сути, само меню и есть. Навигация. Пункты меню могут ссылаться на страницы сайта, категории модулей и элементы модулей, а также на любую внешнюю ссылку.

Если перед названием пункта меню имеется знак +, это значит, что этому пункту принадлежат вложенные пункты меню. Добавить пункт нижнего уровня можно нажав на знак  рядом с названием текущего пункта в общем списке.

Кнопка  служит для удаления текущего пункта меню. Если пункту меню принадлежат пункты нижнего уровня, то при удалении они будут также удалены.

Кнопка  служит для быстрой публикации пункта меню (или, наоборот, блокирования).

Отсортировать пункты меню можно мышкой, просто перетаскивая их нажав на кнопку .

Редактировать

Название

Новости

Показывать на сайте

☒

Ссылка

<http://test.diafan.ru/andrey/news/>

Вложенность: принадлежит

[нет](#)

Категория

Меню верхнее

Сортировка: установить перед

[Новости](#)

Сохранить

Сохранить и выйти

Сохранить и добавить ещё

В

Пункты меню имеют следующие характеристики:

Основные:

- **Название пункта меню** – Это название выводится как пункт меню на сайте. Параметр имеет разные значения для разных языковых версий сайта.
- **Опубликовать на сайте** – Показывать ли пункт меню на сайте или временно скрыть. Параметр имеет разные значения для разных языковых версий сайта.
- **Ссылка** – Ссылка на страницу сайта.
- **Атрибуты ссылки** – HTML-код, выводимый в виде атрибутов для тега ссылки `<a></a>` (Веб-мастеру и программисту).
- **Меню** – Выбор меню, к которому относится текущий пункт.
- **Открывать в новом окне** – Если отмечена, клик пользователя по пункту меню на сайте откроет ссылку в новом окне.
- **Вложенность: принадлежит** – Перемещение текущего пункта меню и всех его подпунктов в принадлежность другому пункту меню (администратору сайта).
- **Сортировка: установить перед** – Изменить положение текущего пункта меню среди других пунктов (администратору сайта).
- **Изображение** – Назначить текущему пункту меню изображение.

При блокировании элемента, на который ссылается пункт меню, соответствующий пункт меню тоже блокируется. Также при удалении элемента – соответствующий пункт меню удаляется. То есть, если скрыть или удалить страницу на сайте, соответствующий пункт меню также будет скрыт или удален.

Структуру меню можно формировать непосредственно из модуля «Меню на сайте» или при заполнении страниц сайта, категорий или элементов модулей, отмечая галку «Показывать в меню». При этом структура сайта повторяется в меню.

Пример:

Отмечаем опцию «Показывать в меню» для новости. Если категория, которой принадлежит новость, также имеет ссылку в меню, то новый пункт меню добавится как

вложенный в пункт категории меню. Иначе добавится пункт меню первого уровня.

При формировании списка элементов, которые можно выбрать в качестве ссылок для пунктов меню, используются элементы других модулей. Ссылки на эти элементы формируются с помощью файла *modules/модуль/admin/модуль.admin.menu.php*.

Меню

Меню

Пункты меню Меню

+ Добавить меню

☐ Меню слева

☐ Меню каталог товаров

Меню верхнее

Категории меню – это различные меню на сайте. Меню на сайте может быть неограниченное множество. Выводятся они на сайте с помощью шаблонного тега [show_block](#).

Кнопка ✕ служит для удаления меню. Кнопка ⏻ служит для быстрой публикации меню (или, наоборот, временного скрытия). При удалении меню удаляются все принадлежащие ему пункты меню.

Редактировать

Название

Номер

Показывать на сайте ☒

Показывать заголовок ☐ ?

Раскрывать все пункты меню ☐ ?

Не отображать ссылку на элемент, если он имеет дочерние пункты ☐

Текущий пункт меню как ссылка ☐ ?

Не отображать имя пункта меню, ☐ ?

Категории меню имеют следующие характеристики:

Основные:

- **Название меню** – Название меню, отображается на сайте перед меню, если отмечен параметр «Показывать заголовок меню». Параметр имеет разные значения для разных языковых версий сайта.
- **Опубликовать на сайте** – Показывать ли меню на сайте. Параметр имеет разные значения для разных языковых версий сайта.
- **Показывать заголовок меню** – Если отмечено, перед пунктами меню выведется название меню (администратору сайта).
- **Раскрывать все пункты меню** – Если отмечено, в меню будут выводиться все пункты меню, включая вложенные. Иначе вложенные пункты будут появляться только при переходе на родительский пункт. (администратору сайта).
- **Не отображать ссылку на элемент, если он имеет дочерние пункты** – Если отмечено, пункты меню не будут ссылками, если у них есть вложенные пункты. (администратору сайта).
- **Текущий пункт меню как ссылка** – Если отмечено, активный пункт меню останется ссылкой. (администратору сайта).
- **Не отображать имя пункта меню, если используется изображение** – Если к пункту меню прикреплено изображение, то имя пункта отображаться не будет. (администратору сайта).
- **Отображать на страницах** – Выбор отдельных страниц сайта, где будет показываться меню. Удерживайте CTRL, чтобы выбрать несколько страниц (администратору сайта).
- **Доступ** – Если отметить опцию «Доступ только», категорию увидят только авторизованные на сайте пользователи, отмеченных типов. Не авторизованные, в том числе поисковые роботы, увидят «404 Страница не найдена» (администратору сайта).
- **Номер** – Номер элемента в БД (Веб-мастеру и программисту).
- **Шаблон вывода меню** – Шаблон будет использован, если в шаблонном теге `show_block` указан атрибут `template="select"`. (Веб-мастеру и программисту. Не меняйте этот параметр, если не уверены в результате!)

Настройки

- **Использовать изображения** – Если отметить, то к каждому пункту меню затем можно будет загружать изображения. Необходимо для графических сайтов, где пункты меню нарисованы, а не написаны текстом
- **Генерировать размеры изображений** – Размеры изображений, заданные в модуле «Изображения» и тег латинскими буквами для подключения изображения на сайте. Обязательно должны быть заданы два размера: превью изображения в списке файлов (тег `medium`) и полное изображение (тег `large`).
- **Максимальный размер загружаемых файлов** – Параметр показывает максимально допустимый размер загружаемых файлов, установленный в настройках хостинга. Параметр выводится, если отмечена опция «Использовать изображения».
- **Применить настройки ко всем ранее загруженным изображениям** – Позволяет перекодировать размер уже загруженных изображений. Кнопка необходима, если изменены настройки размеров изображений. Параметр выводится, если отмечена опция «Использовать изображения».

Шаблонные теги

Для работы с модулем «Меню» служат следующие [шаблонные теги](#):

show_block – выводит меню.

Атрибуты:

- **id** – идентификатор категории меню;
- **template** – имя шаблона;

Внимание!

Атрибут **template** является «переключателем» режима работы шаблонного тега. Если в шаблонном теге есть этот атрибут, то меню на сайте оформляется с помощью традиционных шаблонов модулей. Если этого атрибута нет, то меню на сайте оформляется с помощью атрибутов оформления.

Если в атрибуте **template** указано какое-либо значение, то должен существовать файл *modules/menu/view/menu.view.show_block_***template***.php*.

Пример:

```
<insert name="show_block" module="menu" template="supermenu">
```

меню оформится с помощью файла *modules/menu/view/menu.view.show_block_***supermenu***.php*, который необходимо будет создать (можно скопировать любой другой существующий файл шаблона)

Значение **template="default"** подключит шаблон *modules/menu/view/menu.view.show_block.php*.

Значение **template="select"** подключит шаблон выбранный для меню в административном интерфейсе – параметр «Шаблон вывода меню».

Если атрибут **template не задан**, меню оформляется с помощью атрибутов оформления и выводится шаблоном *modules/menu/view/menu.view.show_menu.php*.

— **tag_start_номер** – текст, выводимый перед каждой ссылкой меню (может содержать слово **Increment** – при выводе автоматически заменяемое на арифметическую прогрессию, 1, 2, 3, ... и **Level** – заменяемое на номер уровня меню);

— **tag_end_номер** – текст, выводимый после каждой ссылки пункта меню (может содержать слово **Increment** - заменяемое на увеличивающийся номер и **Level** – заменяемое на номер уровня меню);

— **tag_level_start_номер** – текст, выводимый перед уровнем (номер) меню;

— **tag_level_end_номер** – текст, выводимый после уровня (номер) меню;

- **tag_active_start_номер** – текст, выводимый перед активным пунктом меню уровня (может содержать слово **Increment** - заменяемое на увеличивающийся номер и **Level** – заменяемое на номер уровня меню);
- **tag_active_end_номер** – текст, выводимый после активного пункта меню уровня (может содержать слово **Increment** - заменяемое на увеличивающийся номер и **Level** – заменяемое на номер уровня меню);
- **tag_active_child_start_номер** – текст, выводимый перед пунктом меню уровня (номер) с активным дочерним пунктом (может содержать слово **Increment** - заменяемое на увеличивающийся номер и **Level** – заменяемое на номер уровня меню);
- **tag_active_child_end_номер** – текст, выводимый после пункта меню уровня (номер) с активным дочерним пунктом (может содержать слово **Increment** - заменяемое на увеличивающийся номер и **Level** – заменяемое на номер уровня меню);
- **separator_номер** – текст, разделяющий пункты меню.

Номер в названии атрибутов обозначает номер уровня. Если атрибуты не указаны для текущего уровня, то они наследуются у верхнего уровня.

Пример:

```
<insert name="show_block" module="menu">
```

выведет просто список ссылок основного меню

```
<table>
  <insert name="show_block" module="menu" id="2" tag_start_1="[tr][td]" tag_end_1="[td]
[/tr]">
</table>
```

сформирует таблицу, в отдельной ячейке которой будут пункты меню из второго меню

```
<insert name="show_block" module="menu" tag_start_1="[div class=`menu`]"
tag_end_1="[div]" tag_start_2="[div class=`submenu`]" tag_end_2="[div]">
```

Выведет:

```
<div class="menu"><a href="...">О нас</a></div>
  <div class="submenu"><a href="...">Животные</a></div>
  <div class="submenu"><a href="...">Миссия</a></div>
  <div class="submenu"><a href="...">История</a></div>
<div class="menu"><a href="...">Фотогалерея</a></div>
```

```
<insert name="show_block" module="menu" tag_start_1="[div class=`menuLevel`]"
```

```
tag_end_1="[/div]">
```

Выведет:

```
<div class="menu1"><a href="...">О нас</a></div>
  <div class="menu2"><a href="...">Животные</a></div>
  <div class="menu2"><a href="...">Миссия</a></div>
  <div class="menu2"><a href="...">История</a></div>
<div class="menu1"><a href="...">Фотогалерея</a></div>
```

Файлы

Модуль находится в папке *modules/menu* и устроен [классически](#).

1. *modules/menu/admin/menu.admin.php* – редактирование пунктов меню;
2. *modules/menu/admin/menu.admin.ajax.php* – обработка Ajax-запросов при работе с меню в административной части;
3. *modules/menu/admin/menu.admin.category.php* – редактирование категорий меню;
4. *modules/menu/admin/menu.admin.config.php* – настройки модуля;
5. *modules/menu/admin/menu.admin.config.js* – настройки модуля, JS-сценарий;
6. *modules/menu/admin/menu.admin.inc.php* – подключение модуля к административной части других модулей;
7. *modules/menu/admin/menu.admin.js* – редактирование пунктов меню, JS-сценарий;
8. *modules/menu/menu.php* – контроллер;
9. *modules/menu/menu.model.php* – модель;
10. *modules/menu/views/menu.view.show_block.php* – шаблон меню, оформленного шаблоном;
11. *modules/menu/views/menu.view.show_block_topmenu.php* – шаблон меню `template=topmenu`;
12. *modules/menu/views/menu.view.show_level.php* – шаблон вывода первого уровня меню, вызывается из функции `show_block` в начале файла, оформленного шаблоном;
13. *modules/menu/views/menu.view.show_level_2.php* – шаблон вывода второго и последующих уровней меню, оформленного шаблоном;
14. *modules/menu/views/menu.view.show_level_topmenu.php* – шаблон вывода первого уровня меню, вызывается из функции `show_block` в начале файла, `template=topmenu`;
15. *modules/menu/views/menu.view.show_level_topmenu_2.php* – шаблон второго и последующих уровней меню, оформленного шаблоном;
16. *modules/menu/views/menu.view.show_menu.php* – шаблон меню, оформленного атрибутами тега.

Пользователи сайта

- [Пользователи](#)
- [Права доступа](#)
- [Регистрация](#)
- [Восстановление доступа](#)
- [Конструктор формы регистрации](#)
- [Настройки](#)
- [Шаблонные теги](#)
- [Файлы](#)

Модуль предназначен для работы с пользователями сайта и их свойствами.

Административная часть модуля относится к части CMS «Управление» и является обязательной для установки.

Пользовательская часть модуля устанавливается опционально и состоит из трех частей: страница пользователя, регистрация, восстановление пароля.

Пользователи

Пользователи сайта

[Пользователи](#) [Права доступа](#) [Конструктор формы регистрации](#)

ФИО: Права доступа:

 [Добавить пользователя](#)

☐	Филимон Авруцкий	filimon
☐	Кукушкин Адольф	kukushkin
	Администратор	admin на сайте Админ

По умолчанию в системе существует главный администратор, имя и пароль которого задаются при установке diafan.CMS.

Пользователь не может удалить, заблокировать самого себя или изменить себе права доступа.

Пользователей можно искать по ФИО.

Кнопка  служит для удаления пользователя. Кнопка  служит для быстрой активации/блокирования пользователя.

Редактировать

ФИО

Филимон Авруцкий ?

Дата регистрации

09.02.2009 ?

Активен

☒ ?

Логин

filimon ?

Пароль

?

Email

filimon@site.ru ?

Пользователи имеют следующие характеристики:

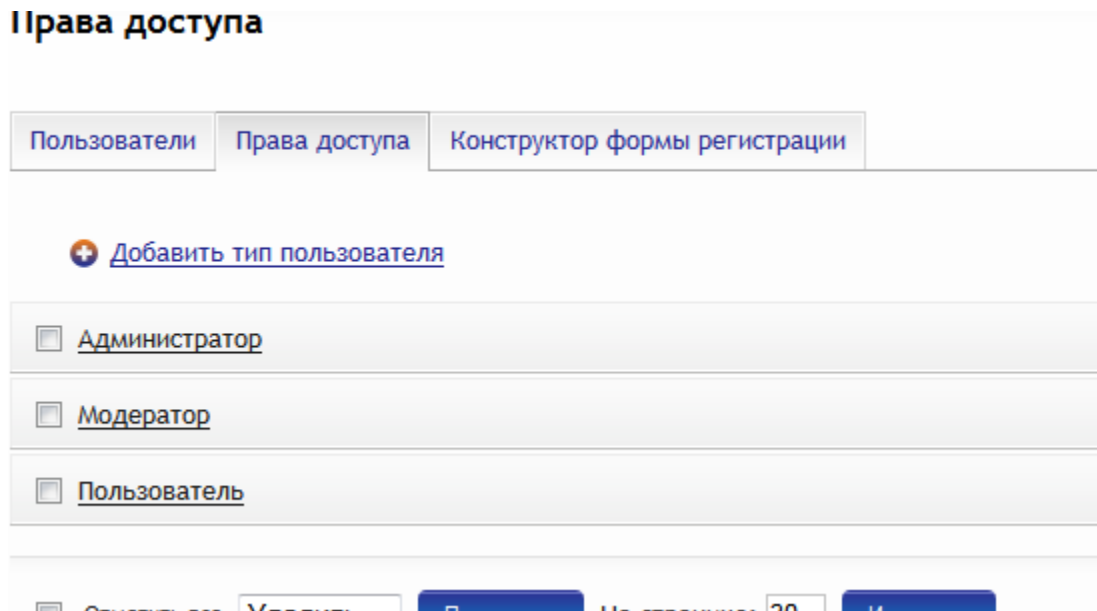
Основные:

- **ФИО** – Свободное информационное текстовое поле.
- **Дата регистрации** – В формате дд.мм.гггг чч:мм, при регистрации устанавливается текущая.
- **Активен** – Если не отмечена, пользователь не сможет авторизоваться на сайте
- **Логин** – Имя пользователя для авторизации на сайте и входа в систему администрирования (если установлен соответствующий тип пользователя). Только цифры и латинские буквы.
- **Пароль** – Пароль для входа в систему администрирования. Только цифры и латинские буквы. Если не указан, пароль не перезаписывается.
- **Email** – Почтовый адрес пользователя в формате mail@site.ru.
- **Телефон** – Телефон для SMS, в федеральном формате.
- **Тип пользователя** – Тип прав пользователя. Уровень доступа настраивается в модуле «Права доступа».
- **Стартовая страница административной части** – Первая страница, загружаемая при авторизации администратора. По умолчанию «Страницы сайта». Возможно установить любую другую, согласно потребностям и тематике сайта, например, «Заказы» или «Вопрос-Ответ».
- **Подключить панель быстрого редактирования** – Если отмечено, в пользовательской части сайта в шапке будет выводиться панель, с помощью которой можно редактировать контент прямо на сайте.
- **Использовать визуальный редактор** – Если отмечено, при редактировании контента во всех модулях будет использоваться визуальный редактор.
- **Сохранять картинки с внешних сайтов, при вставке контента в визуальный редактор** – Если отмечено, при вставке контента в визуальный редактор во всех модулях, будут определяться используемые в контенте изображения (как тег `img src`), сохраняться на сервере и вставляться как локальные. Например, если выделить и скопировать текст с изображениями с новостного сайта, а затем вставить на Ваш сайт, изображения будут автоматически сохранены на Ваш сайт, а ссылки заменены на локальные.
- **URL на страницу в соц. сети** – Заполняется при авторизации через соц. сеть. Служит

идентификатором, с помощью которого пользователь может зайти в аккаунт.

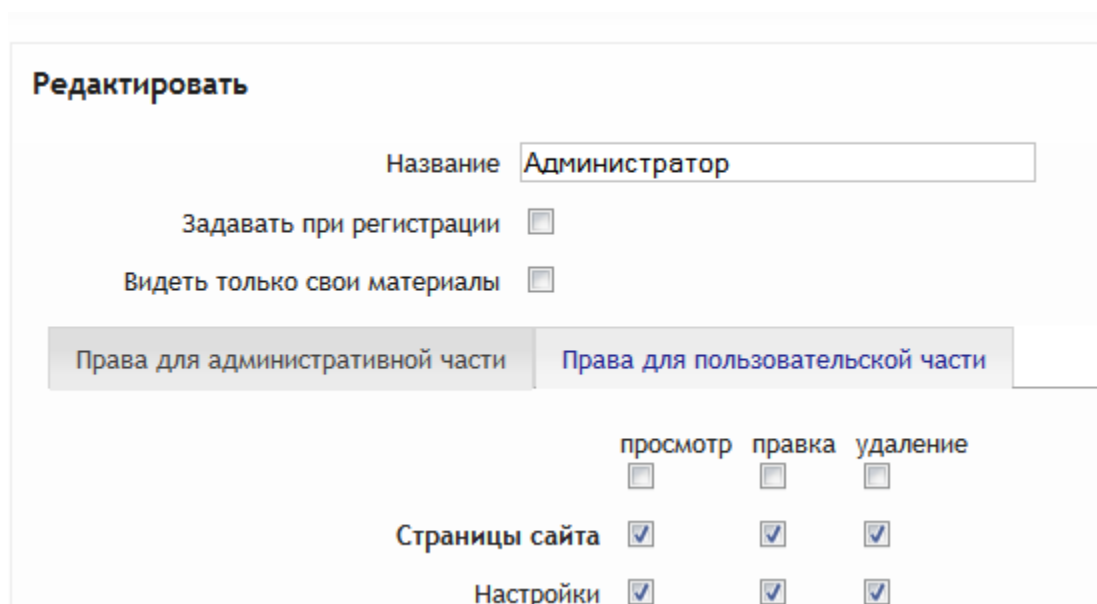
- **Язык интерфейса** – Если предусмотрено несколько языковых версий сайта, то этой опцией можно задать какую версию сайта открыть при авторизации пользователя. При регистрации система запоминает текущую языковую версию.
- **Аватар** – Небольшое изображение для форума, личного кабинета пользователя. Параметр появляется, если в настройках модуля отмечена опция «Использовать аватар».
- **Дополнительные поля** – Группа полей, определенных в части «Конструктор формы регистрации». Поля выводятся соответственно выбранному типу пользователя.

Права доступа



Часть модуля «Права доступа» позволяет редактировать возможности пользователей сайта.

Кнопка ✕ служит для удаления типа пользователя.



	просмотр	правка	удаление
Страницы сайта	☒	☒	☒
Настройки	☒	☒	☒

Типы доступа имеют следующие характеристики:

Основные:

- **Название** – Параметр имеет разные значения для разных языковых версий сайта.
- **Назначать при регистрации на сайте** – Если опция отмечена у одного типа пользователя, то новому пользователю при регистрации на сайте присваивается указанный тип. Если опцией отмечено несколько типов пользователей, то в форме регистрации появляется возможность выбрать тип регистрируемого пользователя. В зависимости от выбранного типа может меняться набор дополнительных полей в форме регистрации.
- **Видеть только свои материалы** – Если опция отмечена, то пользователи указанного типа могут видеть и редактировать только свои материалы в контентных модулях («Страницы сайта», «Новости», «Товары», «Фотографии» и пр.).
- **Привелегии** – Таблица с возможностью разрешения определенного набора действий для каждого модуля в административной части сайта и некоторых, заданных в модулях, действий для пользовательской части сайта. Набор действий модуля для пользовательской части можно задать в файле *modules/модуль/admin/модуль.админ.role.php*.

Формат файла:

Пример:

```
class Модуль_admin_role
{
    public function get_rules()
    {
        $rules = array('название_латинскими_буквами' => 'Название на русском');
        return $rules;
    }
}
```

Регистрация

Модуль «Регистрация» позволяет регистрироваться на сайте пользователям для получения дополнительных возможностей.

Чтобы вывести форму регистрации / редактирования на сайте, нужно прикрепить модуль к странице сайта.

Регистрация и редактирование осуществляются с помощью технологии Ajax, то есть без перезагрузки всей страницы.

Восстановление доступа

Восстановление доступа – это отдельный модуль в пользовательской части сайта (*reminding*). Он устанавливается вместе с модулем «Регистрация» и необходим для восстановления пароля пользователями.

Чтобы вывести форму восстановления доступа на сайте, нужно прикрепить модуль к странице сайта.

Восстановление доступа осуществляются с помощью технологии Ajax, то есть без перезагрузки всей страницы.

Конструктор формы регистрации

Конструктор формы регистрации

The screenshot shows the 'Конструктор формы регистрации' (Registration Form Builder) tab. At the top, there are three tabs: 'Пользователи' (Users), 'Права доступа' (Permissions), and 'Конструктор формы регистрации' (Registration Form Builder). Below the tabs, there is a '+ Добавить поле' (Add field) button. A list of fields is shown, with one field named 'Дополнительные сведения' (Additional information) selected. At the bottom, there are controls for the list: a checkbox for 'Отметить все' (Select all), a 'Удалить' (Delete) button, a 'Применить' (Apply) button, a 'На странице: 30' (Per page: 30) label, and an 'Изменить' (Change) button.

Форму регистрации / редактирования данных можно дополнить своими полями с помощью конструктора формы.

Порядок отображения полей можно редактировать с помощью мыши и кнопки . Кнопка служит для удаления полей.

Конструктор – это дополнительные поля формы регистрации.

The screenshot shows the 'Добавить новый' (Add new) dialog for configuring a new field. It has the following fields and options:

- Название** (Name):
- Обязательно для заполнения** (Required for filling): ☐
- Выводить на странице пользователя** (Display on user page): ☒
- Выводить в форме регистрации** (Display in registration form): ☒
- Выводить в форме редактирования данных** (Display in data editing form): ☒
- Тип пользователя** (User type):
 - Пользователь
 - Модератор
 - Администратор

Дополнительные поля имеют следующие параметры.

Основные:

- **Название** – Параметр имеет разные значения для разных языковых версий сайта.
- **Обязательно для заполнения.**
- **Выводить на странице пользователя.**
- **Выводить в форме регистрации.**
- **Выводить в форме редактирования данных.**
- **Тип пользователя** – Поле прикрепляется к одному или нескольким типам пользователей. При редактировании и в форме регистрации при смене типа пользователя меняется набор полей.
- **Сортировка: установить перед** – Редактирование порядка следования поля в списке полей.
- **Тип** – Возможные значения:
 - строка,
 - число,
 - дата,
 - дата и время,
 - текстовое поле,
 - галочка,
 - выпадающий список,
 - список с выбором нескольких значений,
 - электронный ящик,
 - заголовок группы характеристик,
 - файлы,
 - изображения.
- **Максимальное количество добавляемых файлов** – Количество добавляемых файлов. Если значение равно нулю, то форма добавления файлов не выводится. Параметр выводится, если тип характеристики задан как «файлы».
- **Доступные типы файлов (через запятую)** – Параметр выводится, если тип характеристики задан как «файлы».

Пример:

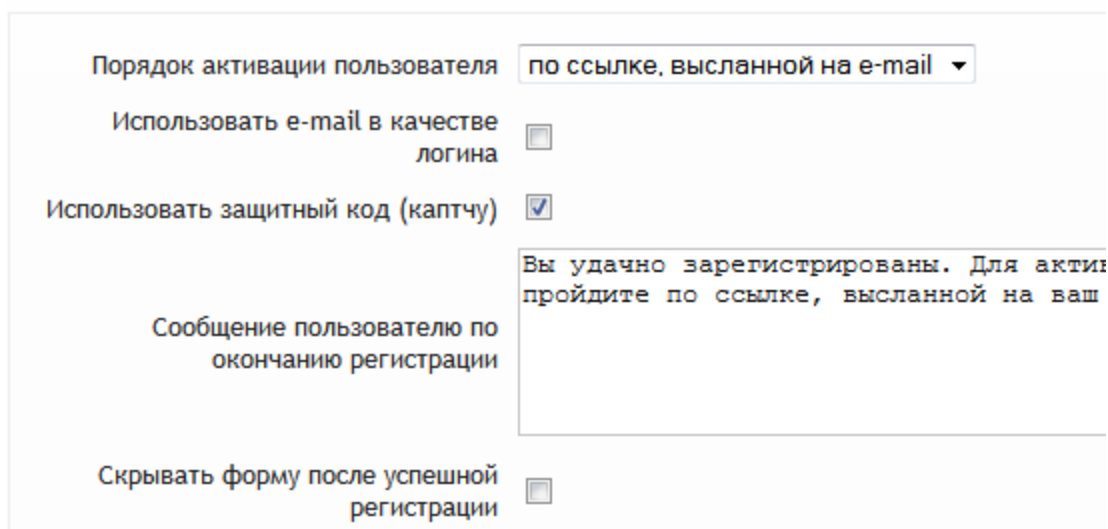
doc, gif, jpg, mpg, pdf, png, txt, zip

- **Распознавать изображения** – Позволяет прикрепленные файлы в формате JPEG, GIF, PNG отображать как изображения. Параметр выводится, если тип характеристики задан как «файлы».
- **Размер для большого изображения** – Размер изображения, отображаемый в пользовательской части сайта при увеличении изображения предпросмотра. Параметр выводится, если тип характеристики задан как «файлы» и отмечена опция «Распознавать изображения».
- **Размер для маленького изображения** – Размер изображения предпросмотра. Параметр выводится, если тип характеристики задан как «файлы» и отмечена опция «Распознавать изображения».
- **Использовать анимацию при увеличении изображений** – Параметр добавляет JavaScript код, позволяющий включить анимацию при увеличении изображений. Параметр выводится, если отмечена опция «Распознавать изображения». Параметр выводится, если тип характеристики задан как «файлы» и отмечена опция «Распознавать изображения».
- **Максимальный размер загружаемых файлов** – Параметр показывает максимально допустимый размер загружаемых файлов, установленный в настройках хостинга. Параметр выводится, если тип характеристики задан как «файлы».

- **Генерировать размеры изображений** – Размеры изображений, заданные в модуле «Изображения». Параметр выводится, если тип характеристики задан как «изображение».
- **Значения** – Появляется для полей с типом «галочка», «выпадающий список» и «список с выбором нескольких значений».

Настройки

Настройки



Порядок активации пользователя: по ссылке, высланной на e-mail ▼

Использовать e-mail в качестве логина: ☐

Использовать защитный код (капчу): ☒

Сообщение пользователю по окончании регистрации: Вы успешно зарегистрированы. Для активации аккаунта перейдите по ссылке, высланной на ваш e-mail.

Скрывать форму после успешной регистрации: ☐

- **Порядок активации пользователя** – Возможные значения:
 - при регистрации,
 - по ссылке, высланной на e-mail,
 - администратором.
- **Использовать e-mail в качестве логина** – Позволяет отказаться от поля «Логин» в форме регистрации, редактирования данных и при авторизации в качестве логина использовать e-mail.
- **Использовать защитный код (капчу)** – При регистрации пользователь должен ввести защитный код.
- **Сообщение пользователю по окончании регистрации** – Параметр имеет разные значения для разных языковых версий сайта.

Пример:

Вы успешно зарегистрированы. Для активации аккаунта перейдите по ссылке, высланной на ваш e-mail.

- **Скрывать форму после успешной регистрации** – Позволяет скрывать форму после успешной регистрации.
- **Использовать авторизацию через сервис Loginza** – Подключает авторизацию через социальные сети.
- **ID виджета для сервиса Loginza** – Данные из настроек сервиса Loginza. Параметр появляется, если отмечена опция «Использовать авторизацию через сервис Loginza».
- **Секретный ключ для сервиса Loginza** – Данные из настроек сервиса Loginza. Параметр появляется, если отмечена опция «Использовать авторизацию через сервис Loginza».

- **Формат даты** – Позволяет настроить отображение даты в модуле. Возможные значения:
 - 01.05.2013,
 - 01.05.2013 14:45,
 - 1 мая 2013 г.,
 - 1 мая,
 - 1 мая 2013, понедельник,
 - вчера 15:30,
 - не отображать.
- **Использовать аватар** – Подключает аватар к пользователям сайта. Аватар загружается при регистрации, редактировании данных пользователем, редактировании данных о пользователе администратором. Используется аватар на форуме, в комментариях и выводятся вместе с приветствием пользователю.
- **Размер аватара** – Аватар будет генерироваться строго заданного размера. Параметр появляется, если отмечена опция «Использовать аватар».
- **Уведомлять администратора** – Возможность уведомления администратора о регистрации новых пользователей.
- **E-mail для уведомлений администратора** – Возможные значения:
 - e-mail, указанный в параметрах сайта;
 - другой (при выборе этого значения появляется дополнительное поле **впишите e-mail**).

Параметр появляется, если отмечена опция «Уведомлять администратора».

- **Тема письма администратору** – Можно добавлять:
 - %title – название сайта,
 - %url – адрес сайта (например, site.ru).

Параметр появляется, если отмечена опция «Уведомлять администратора».

Пример:

%title (%url). Новый пользователь

- **Текст письма администратору** – Можно добавлять:
 - %title – название сайта,
 - %url – адрес сайта (например, site.ru),
 - %fio – ФИО пользователя,
 - %login – логин пользователя,
 - %email – e-mail пользователя.

Параметр появляется, если отмечена опция «Уведомлять администратора».

Пример:

Здравствуйте, администратор сайта *%title (%url)!*
На сайте появился новый пользователь: *%fio (%login), %email*

- **E-mail, указываемый в обратном адресе пользователю** – Возможные значения:
 - e-mail, указанный в параметрах сайта;
 - другой (при выборе этого значения появляется дополнительное поле **впишите e-mail**).
- **Тема письма новому пользователю** – Можно добавлять:
 - %title – название сайта,
 - %url – адрес сайта (например, site.ru).

Параметр имеет разные значения для разных языковых версий сайта.

Пример:

Вы зарегистрированы на сайте %title (%url)

• **Сообщение новому пользователю** – Можно добавлять:

- %title – название сайта,
- %url – адрес сайта (например, site.ru),
- %fio – ФИО пользователя,
- %login – логин пользователя,
- %password – пароль пользователя,
- %email – e-mail пользователя,
- %actlink – ссылка для активации аккаунта.

Параметр имеет разные значения для разных языковых версий сайта.

Пример:

если активация происходит при регистрации:

Здравствуйте, %fio!
 Вы зарегистрированы на сайте %title (%url).
 Логин: %login
 Пароль: %password

если активация происходит по ссылке:

Здравствуйте, %fio!
 Вы зарегистрированы на сайте %title (%url).
 Логин: %login
 Пароль: %password
 Для активации аккаунта пройдите по ссылке %actlink
 Ссылка действует в течении суток.

если активацию производит администратор:

Здравствуйте, %fio!
 Вы зарегистрированы на сайте %title (%url).
 Логин: %login
 Пароль: %password
 Ваш аккаунт будет активирован после проверки администратором.

• **Сообщение пользователю при восстановлении пароля** – Параметр имеет разные значения для разных языковых версий сайта.

Пример:

На ваш e-mail отправлена ссылка на форму изменения пароля.

• **Тема письма со ссылкой на изменение пароля** – Можно добавлять:

- %title – название сайта,
- %url – адрес сайта (например, site.ru).

Параметр имеет разные значения для разных языковых версий сайта.

Пример:

Восстановление доступа к сайту %title (%url)

- **Текст письма со ссылкой на изменение пароля** – Можно добавлять:
 - %title – название сайта,
 - %url – адрес сайта (например, site.ru),
 - %actlink – ссылка на изменение пароль.

Параметр имеет разные значения для разных языковых версий сайта.

Пример:

Здравствуйте, %fio! Вы запросили восстановление доступа к сайту %title (%url). Для изменения пароля пройдите по ссылке: %actlink

- **Тема письма с новым паролем** – Можно добавлять:
 - %title – название сайта,
 - %url – адрес сайта (например, site.ru).

Параметр имеет разные значения для разных языковых версий сайта.

Пример:

Новый пароль на сайте %title (%url)

- **Текст письма с новым паролем** – Можно добавлять:
 - %title – название сайта,
 - %url – адрес сайта (например, site.ru),
 - %fio – ФИО пользователя,
 - %login – логин пользователя,
 - %password – новый пароль пользователя.

Параметр имеет разные значения для разных языковых версий сайта.

Пример:

Здравствуйте, %fio! Вы изменили пароль на сайте %title (%url). Логин: %login
Пароль: %password

Шаблонные теги

Для работы с модулем «Пользователи» служат следующие [шаблонные теги](#):

show_block – выводит статистику пользователей на сайте.

Атрибуты:

— **template** – шаблон тега (файл `modules/users/views/users.view.show_block_template.php`; по умолчанию шаблон `modules/users/views/users.view.show_block.php`).

Пример:

```
<insert name="show_block" module="users">
```

выведет «Сейчас на сайте: 13 гостей, 9 пользователей.»

show_login module="registration" – выводит форму авторизации или приветствие и ссылки

на редактирование данных и выход, если пользователь авторизован.

Атрибуты:

— **template** – шаблон тега (файл *modules/registration/views/registration.view.show_login_***template***.php*; по умолчанию шаблон *modules/registration/views/registration.view.show_login.php*).

Пример:

```
<insert name="show_login" module="registration">
```

выведет форму авторизации

Файлы

1. *modules/registration/registration.php* – контроллер модуля «Регистрация»;
2. *modules/registration/registration.ajax.php* – обработка запроса при регистрации пользователя;
3. *modules/registration/registration.js* – JS-сценарий модуля «Регистрация»;
4. *modules/registration/registration.model.php* – модель модуля «Регистрация»;
5. *modules/registration/views/registration.view.act.php* – шаблон страницы активации аккаунта;
6. *modules/registration/views/registration.view.form.php* – шаблон формы регистрации;
7. *modules/registration/views/registration.view.images.php* – шаблон прикрепленных изображений;
8. *modules/registration/views/registration.view.show_login.php* – шаблон блока авторизации;
9. *modules/registration/views/registration.view.show_param.php* – шаблон дополнительных полей в форме регистрации данных;
10. *modules/registration/views/registration.view.success.php* – шаблон страницы успешной регистрации;
11. *modules/reminding/reminding.php* – контроллер модуля «Восстановление пароля»;
12. *modules/reminding/reminding.ajax.php* – обработка запроса при отправке данных из формы восстановления пароля;
13. *modules/reminding/reminding.model.php* – модель модуля «Восстановление пароля»;
14. *modules/reminding/views/reminding.view.form_change_password.php* – шаблон формы смены пароля;
15. *modules/reminding/views/reminding.view.form_mail.php* – шаблон формы восстановления доступа;

16. *modules/reminding/views/reminding.view.success.php* – шаблон страницы успешной смены пароля;
17. *modules/userpage/userpage.php* – контроллер модуля «Страница пользователя»;
18. *modules/userpage/userpage.model.php* – модель модуля «Страница пользователя»;
19. *modules/userpage/views/userpage.view.orders.php* – шаблон заказов пользователя;
20. *modules/userpage/views/userpage.view.show.php* – шаблон страницы пользователя;
21. *modules/users/admin/users.admin.php* – редактирование пользователей сайта;
22. *modules/users/admin/users.admin.config.php* – настройки модуля;
23. *modules/users/admin/users.admin.param.php* – конструктор формы регистрации;
24. *modules/users/admin/users.admin.role.php* – редактирование типов пользователей;
25. *modules/users/admin/users.admin.role.js* – редактирование типов пользователей, JS-сценарий;
26. *modules/users/users.php* – котроллер модуля «Пользователи на сайте»;
27. *modules/users/users.install.php* – установка модуля;
28. *modules/users/users.model.php* – модель модуля «Пользователи на сайте»;
29. *modules/users/views/users.view.show_block.php* – шаблон статистики пользователей на сайте;
30. *modules/usersettings/usersettings.php* – контроллер модуля «Настройки аккаунта»;
31. *modules/usersettings/usersettings.ajax.php* – обработка запроса при изменении данных о пользователе;
32. *modules/usersettings/usersettings.js* – JS-сценарий модуля «Настройки аккаунта»;
33. *modules/usersettings/usersettings.model.php* – модель модуля «Настройки аккаунта»;
34. *modules/usersettings/views/usersettings.view.attachments.php* – шаблон прикрепленных файлов в настройках аккаунта;
35. *modules/usersettings/views/usersettings.view.avatar.php* – шаблон аватара;
36. *modules/usersettings/views/usersettings.view.form.php* – шаблон формы настроек аккаунта;
37. *modules/usersettings/views/usersettings.view.images.php* – шаблон прикрепленных изображений;
38. *modules/usersettings/views/usersettings.view.show_param.php* – шаблон дополнительных в настройках аккаунта.

Обновление CMS

- [Автообновление](#)
- [Список закрытых для обновления файлов](#)
- [Файлы](#)

Модуль относится к части CMS «Управление», является частью ядра diafan.CMS и обязателен для установки. Имеет только административную часть.

Автообновление

Модуль позволяет обновляться в пределах [минорной версии](#) diafan.CMS. Загрузка обновленных файлов производится по протоколу FTP, поэтому необходимо указать параметры подключения в [«Параметрах сайта»](#).

Обновление доступно только для сайтов, на которые куплена лицензия.

Обновление CMS

Автообновление	Список закрытых для обновления файлов
--------------------------------	---

Укажите параметры подключения к FTP-серверу вашего сайта.
[Параметры сайта](#)

Список закрытых для обновления файлов

Если вы вносили изменения в файлы, доступные к обновлению, вы можете поместить их в специальный список, чтобы при обновлении случайно их не перезаписать.

Файлы, занесенные в список, будут помечены сообщением «Внимание! Вы изменили файл. Его обновление приведет к потере внесенных изменений». Такой файл можно обновить, игнорируя предупреждение.

Записи имеют следующие характеристики.

Основные:

- **Название** – Путь до файла от корня сайта. Например, *modules/news/news.php*.

- **Пользователь** – Пользователь, добавивший файл в список.
- **Дата** – Вводится в формате дд.мм.гггг чч:мм.
- **Примечание.**

Файлы

1. *modules/update/admin/update.admin.php* – обновление CMS;
2. *modules/update/admin/update.admin.list.php* – список закрытых для обновления файлов.

Модули и БД

- [Установка модулей](#)
- [Восстановление базы данных](#)
- [Экспорт/импорт базы данных](#)
- [Файлы](#)

Модуль относится к части CMS «Настройки», является частью ядра diafan.CMS и обязателен для установки. Имеет только административную часть.

Установка модулей

Модули и БД

[Установка модулей](#)[Восстановление БД](#)[Экспорт/импорт БД](#)

Снимите галку с модулей, которые хотите удалить и поставьте напротив тех, что хотите

☐ Заполнить сайт примерами (может занять время). Только для устанавливаемых модулей.

Доступны следующие модули для установки/удаления:

- ☒ Новости
- ☒ Статьи
- ☒ Обратная связь
- ☒ Вопрос-Ответ

Пользователь может выбрать из списка модули, которые хочет установить или удалить. Если пользователь снимает галочку с уже установленного модуля и нажимает кнопку «Установить», то модуль деинсталлируется. При этом вся информация, заполненная в этом модуле, удаляется. Установить/удалить можно только те модули, которые имеют файл инсталляции: `modules/модуль/модуль.install.php`.

Восстановление базы данных

Восстановление БД

Установка модулей

Восстановление БД

Экспорт/импорт БД

[Начать восстановление базы данных](#)

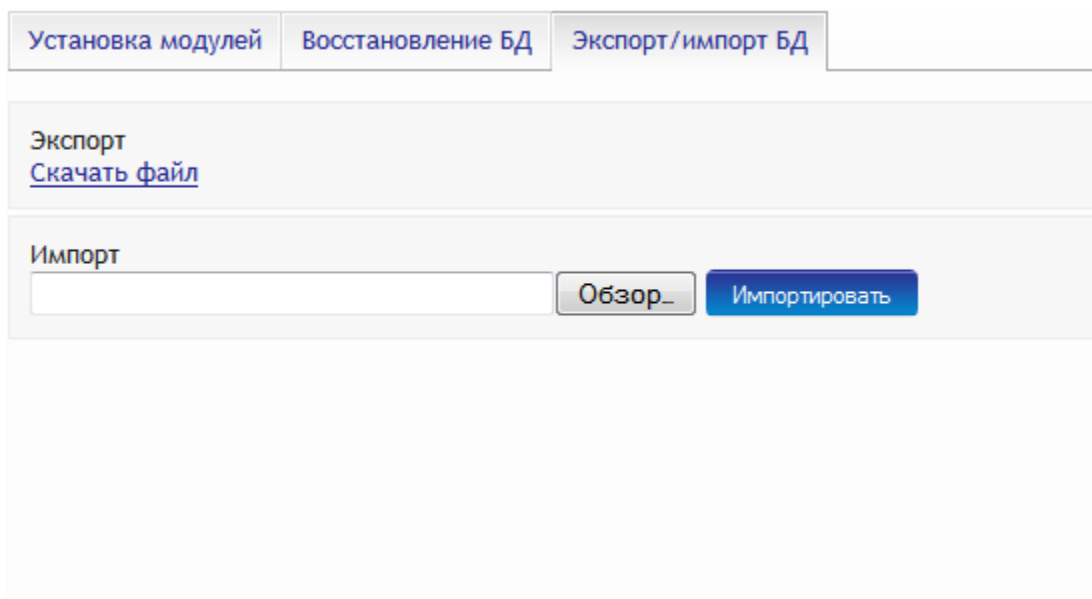
Если обновление на новую минорную версию произведено некорректно или внесены непоправимые изменения в структуру таблиц базы данных, можно воспользоваться инструментом «Восстановление базы данных».

Модуль сравнивает фактические таблицы в базе данных сайта и устанавливаемые при инсталляции. Если в фактических нет необходимых полей или таблиц, они устанавливаются. Лог изменений выводится на экран.

Восстановление базы данных может понадобиться после автообновления. Если релиз включает новую функциональность, затрагивающую базу данных, то после обновления на экран выведется сообщение «Требуется обновление базы данных».

Структура базы данных описана в файлах *installation/core.php* и *modules/модуль/модуль.install.php*.

Экспорт/импорт базы данных



Установка модулей Восстановление БД Экспорт/импорт БД

Экспорт
[Скачать файл](#)

Импорт
 Обзор... Импортировать

Модуль позволяет работать с базой данных напрямую. Экспорт базы данных позволяет скачать текущую базу сайта. Импорт базы данных позволяет выполнить SQL-запросы, отписанные в файлах. С помощью модуля можно выполнять бэкап базы данных. Для восстановления данных из бэкапа просто загрузите ранее сохраненный файл.

Если на сервере подключено расширение PHP [ZipArchive](#), то модуль экспорта будет разбивать файл экспорта на файлы размером 1Mb и сохранять их в Zip-архив. Также при импорте возможно использовать Zip-архив SQL-файлов.

Файлы

1. *includes/install.php* – установка модулей;
2. *modules/update/admin/update.admin.importexport.php* – импорт/экспорт базы данных;
3. *modules/update/admin/update.admin.install.php* – установка/удаление модулей;
4. *modules/update/admin/update.admin.repair.php* – восстановление базы данных;
5. *modules/update/update.export.php* – экспорт БД.

Параметры сайта

- [Файлы](#)

Модуль «Параметры сайта» позволяет редактировать общие настройки сайта. Модуль относится к части CMS «Управление», является частью ядра diafan.CMS и обязателен для установки.

Модуль имеет только административную часть.

Настроить сайт можно по следующим параметрам:

Основные

- **Название сайта** – Если на сайте используются несколько языковых версий, то поле «Название сайта» нужно заполнять для каждой версии.
- **Host для базы данных** – Хост для подключения к базе данных. Например, localhost. Данные обычно предоставляются хостингом при регистрации. (Веб-мастеру и программисту. Не меняйте этот параметр, если не уверены в результате!)
- **База данных** – Название базы данных. Данные обычно предоставляются хостингом при регистрации или создается база данных в панели управления хостингом. При создании базы данных рекомендуется выбирать кодировку UTF8. (Веб-мастеру и программисту. Не меняйте этот параметр, если не уверены в результате!)
- **Пользователь базы данных** – Данные обычно предоставляются хостингом при регистрации или создается база данных в панели управления хостингом. (Веб-мастеру и программисту. Не меняйте этот параметр, если не уверены в результате!)
- **Пароль для базы данных** – Данные обычно предоставляются хостингом при регистрации или создается база данных в панели управления хостингом. (Веб-мастеру и программисту. Не меняйте этот параметр, если не уверены в результате!)
- **Префикс (например, diafan_)** – Символы, добавляемые к каждой таблице в базе данных, используемой CMS. Полезно, когда в одной базе данных MySQL имеются таблицы не только CMS. Префикс может быть пустым. (Веб-мастеру и программисту. Не меняйте этот параметр, если не уверены в результате!)
- **Кодировка базы данных** – diafan.CMS работает с базой данных в кодировке UTF8. Изменить параметр можно в случае индивидуальной настройки системы. (Веб-мастеру и программисту. Не меняйте этот параметр, если не уверены в результате!)
- **Папка для хранения пользовательских файлов.** – Имя папки, где будут храниться все загружаемые файлы для контента сайта. По-умолчанию все пользовательские файлы хранятся в папке *userfiles*. (Веб-мастеру и программисту. Не меняйте этот параметр, если не уверены в результате!)
- **Папка административной части** – Адрес административной части сайта. Например, *http://site.ru/admin/* или *http://site.ru/manager/*. Изменение параметра означает изменение URL-адреса панели администрирования. (Веб-мастеру и программисту. Не меняйте этот параметр, если не уверены в результате!)
- **Разрешить мобильную версию сайта** – Если отмечено, то CMS будет автоматически определять устройство, с которого зашли на сайт и если это мобильное устройство, то автоматически будет загружаться упрощенный шаблон дизайна *themes/m/site.php*.
- **Разрешить загружать сайт во frame** – Если отмечено, то сайт можно будет

вставлять во frame. Не рекомендуется, т.к. повышается опасность clickjacking-атак на сайт.

- **Вариант генерации ЧПУ** – Способ преобразования названия при автоматическом генерировании ЧПУ. Возможные значения:
 - транслит,
 - перевод на английский,
 - русская кириллица.
- **Способ преобразования** – Массив кириллических символов и соответствующих им латинских символов. Символы разделены пробелом. Параметр появляется, если в качестве варианта генерации ЧПУ выбран «транслит».

Значения по-умолчанию:

Пример:

|а|б|в|г|д|е|ё|ж|з|и|й|к|л|м|н|о|п|р|с|т|у|ф|х|ц|ч|ш|щ|ы|э|ю|я|А|Б|В|Г|Д|Е|Ё|Ж|З|И|Й|К|Л|М|Н|О|П|Р|С|Т|У|Ф|Х|Ц|Ч|Ш|Щ|Ы|Э|Ю|Я

Пример:

-|а|б|в|г|д|е|ё|ж|з|и|й|к|л|м|н|о|п|р|с|т|у|ф|х|ц|ч|ш|щ|ы|э|ю|я|А|Б|В|Г|Д|Е|Ё|Ж|З|И|Й|К|Л|М|Н|О|П|Р|С|Т|У|Ф|Х|Ц|Ч|Ш|Щ|Ы|Э|Ю|Я

- **ЧПУ оканчивается на** – Можно использовать слеш или иное окончание. Например, если установить *.php*, все адреса страниц сайта будут формироваться как *http://site.ru/news.php* Для *.html* – *http://site.ru/news.html*. По умолчанию слеш и *http://site.ru/news/*.
- **Генерировать ЧПУ для модулей автоматически** – Формирование ЧПУ для модулей (новостей, категорий новостей, товаров, статей и пр.) в автоматическом режиме из названий. Если галка отключена, ЧПУ отдельного товара будет генерироваться как *http://site.ru/shop/cat1/show5/*. Если галка стоит, то при сохранении ЧПУ сгенерируется автоматически из названия категорий и имени элементов, т.е. *http://site.ru/shop/telefony/nokia8800/*.
- **FTP-хост** – Адрес FTP-сервера, для подключения к хостингу. Используется для доступа к файлам сайта, если не хватает прав доступа. В том числе может быть использовано для автообновления. (Веб-мастеру и программисту. Не меняйте этот параметр, если не уверены в результате!)
- **FTP-логин** – Имя ftp-пользователя, для подключения хостингу. Используется для доступа к файлам сайта, если не хватает прав доступа. В том числе может быть использовано для автообновления. (Веб-мастеру и программисту. Не меняйте этот параметр, если не уверены в результате!)
- **FTP-пароль** – Пароль ftp-пользователя, для подключения хостингу. Используется для доступа к файлам сайта, если не хватает прав доступа. В том числе может быть использовано для автообновления. (Веб-мастеру и программисту. Не меняйте этот параметр, если не уверены в результате!)
- **Относительный путь до сайта** – Нужен, если указанный FTP-пользователь после авторизации попадает не в корень сайта, а несколькими уровнями выше. Тогда нужно указать путь к корню сайта. Например, */www/site.ru/*, узнайте на хостинге. Используется для доступа к файлам сайта, если не хватает прав доступа. В том числе может быть использовано для автообновления. (Веб-мастеру и программисту. Не меняйте этот параметр, если не уверены в результате!)
- **e-mail сайта** – Адрес, на который по-умолчанию приходят уведомления и который

указывается в обратном адресе всех писем. Здесь может быть указан только один адрес.

- **Использовать SMTP-авторизацию при отправке почты с сайта** – Рекомендуется использовать исходящую SMTP-авторизацию, иначе письма-уведомления с сайта могут блокироваться большинством спам-фильтров. Также SMTP-авторизацию нужна, если невозможно отправить письма обычной функцией [mail\(\)](#).
- **SMTP-хост (например, smtp.mail.ru)** – Параметр появляется, если отмечена опция «Использовать SMTP-авторизацию при отправке почты с сайта».
- **SMTP-логин** – Ваш почтовый логин, для входа в почту. Параметр появляется, если отмечена опция «Использовать SMTP-авторизацию при отправке почты с сайта».
- **SMTP-пароль** – Ваш почтовый пароль, для входа в почту. Параметр появляется, если отмечена опция «Использовать SMTP-авторизацию при отправке почты с сайта».
- **SMTP-порт (по умолчанию 25)** – В большинстве случаев можно не указывать. Параметр появляется, если отмечена опция «Использовать SMTP-авторизацию при отправке почты с сайта».
- **Кэширование Memcached** – Подключает Memcached-кэширование. По-умолчанию используется файловое кэширование. Веб-мастеру и программисту. Не меняйте этот параметр, если не уверены в результате!
- **Хост сервера Memcached** – Веб-мастеру и программисту. Не меняйте этот параметр, если не уверены в результате! Параметр появляется, если отмечена опция «Кэширование Memcached».
- **Порт сервера Memcached** – Веб-мастеру и программисту. Не меняйте этот параметр, если не уверены в результате! Параметр появляется, если отмечена опция «Кэширование Memcached».
- **Подключить SMS-уведомления (требуется регистрация).** – SMS-рассылки интегрирована в модули «Обратная связь», «Оформление заказа», «Комментарии», «Вопрос-Ответ» для уведомления администраторов. А также в модуль «Рассылки» для массовой рассылки SMS. Подключить SMS-уведомления нужно в настройках соответствующего модуля. Для включения SMS на сайте необходимо зарегистрироваться в системе [Byte Hand](#).
- **Ключ** – Данные из настроек сервиса Byte Hand. Параметр появляется, если отмечена опция «Подключить SMS-уведомления(требуется регистрация).».
- **ID** – Данные из настроек сервиса Byte Hand. Параметр появляется, если отмечена опция «Подключить SMS-уведомления(требуется регистрация).».
- **Подпись** – Данные из настроек сервиса Byte Hand. Параметр появляется, если отмечена опция «Подключить SMS-уведомления(требуется регистрация).».
- **Таймзона** – Часовой пояс, [список часовых поясов](#). По умолчанию: Europe/Moscow

Режим разработки

- **Включить режим разработки** – Если отметить, в подвале всех страниц сайта будет выводиться консоль, содержащая все уведомления сервера с замечаниями и PHP-ошибками. (Веб-мастеру и программисту. Не меняйте этот параметр, если не уверены в результате!)
- **Перевести сайт в режим обслуживания** – Если отметить, сайт будет доступен только авторизованному администратору. Все остальные посетители сайта будут видеть только страницу *themes/503.php* – «Сайт в разработке, временно недоступен». (Веб-мастеру и программисту. Не меняйте этот параметр, если не уверены в результате!)
- **Отключить кэширование** – Данный параметр разработчику необходимо обязательно

включать при доработке скриптов и обязательно отключать в штатном режиме работы сайта. Постоянно отключенное кэширование может замедлить работу системы! (Веб-мастеру и программисту. Не меняйте этот параметр, если не уверены в результате!)

- **Сбросить кэш** – Если отметить, внутренний кэш сайта будет удален. Галка при этом не останется отмечена. Рекомендуется сбрасывать кеш, после внесения изменений в скрипты. (Веб-мастеру и программисту. Не меняйте этот параметр, если не уверены в результате!)
- **Включить профилирование SQL-запросов** – Если отметить, в подвале всех страниц сайта будет выводиться консоль, содержащая список всех использованных системой SQL-запросов и время их выполнения. (Веб-мастеру и программисту. Не меняйте этот параметр, если не уверены в результате!)

Редактируемые модулем настройки записываются в файл [config.php](#). Для корректной работы с модулем и для редактирования данных необходимо, чтобы этот файл был доступен для записи (права доступа 0777).

Внимание! Ошибочно введенные параметры доступа к базе данных приведут к остановке работы всего сайта. Если это произошло, необходимо вручную редактировать файл *config.php*, подключившись к сайту по FTP.

Этот файл должен храниться в кодировке UTF-8 с параметром without BOM (не использовать служебный символ *Byte Order Mark*), для редактирования используйте редактор, позволяющий изменять этот параметр, например, Notepad++.

Файлы

1. *config.php* – файл конфигурации;
2. *includes/config.php* – сохранение параметров сайта;
3. *modules/config/admin/config.admin.php* – редактирование параметров сайта;
4. *modules/config/admin/config.admin.js* – редактирование параметров сайта, JS-сценарий.

Языки сайта

- [Языки сайта](#)
- [Перевод интерфейса](#)
- [Подключение](#)
- [Шаблонные теги](#)
- [Файлы](#)

Модуль относится к части CMS «Управление», пересекается со всеми модулями сайта и является обязательным для установки.

Языки сайта

The screenshot shows the 'Языки сайта' (Site Languages) management interface. At the top, there are two tabs: 'Языки сайта' (selected) and 'Перевод интерфейса' (Interface Translation). Below the tabs is a '+ Добавить язык сайта' (Add site language) button. A table lists the current languages: 'eng' with an unchecked checkbox, and 'ru' with a checked checkbox and the label 'Основной язык' (Main language). At the bottom, there is a control bar with an 'Отметить все' (Select all) checkbox, a 'Удалить' (Delete) dropdown button, a 'Применить' (Apply) button, a 'На странице: 30' (Items per page: 30) display, and an 'Изменить' (Change) button.

Языки сайта имеют следующие характеристики:

Основные:

- **Язык.**
- **Обозначение языка латиницей** – Используется для формирования URL. Если указан ru или rus, то интерфейс считается русским и не переводится.
- **Основной язык для пользовательской части** – Выберите этот параметр, если хотите изменить язык по умолчанию в пользовательской части
- **Основной язык для административной части** – Выберите этот параметр, если хотите изменить язык по умолчанию в административной части

При добавлении и удалении языка сайта меняется структура базы данных, структура сайта. Нельзя удалить основной язык сайта.

Любой язык сайта можно определить как основной для административной части и основной для пользовательской части. При этом язык, ранее помеченный как основной для соответствующей части, становится не основным. Основным язык нельзя удалить.

Перевод интерфейса

Перевод интерфейса

The screenshot shows the 'Перевод интерфейса' (Interface Translation) module interface. At the top, there are two tabs: 'Языки сайта' (Site Languages) and 'Перевод интерфейса' (Interface Translation), with the latter being the active tab. Below the tabs is a link 'Импорт / экспорт перевода' (Import / export translation). At the bottom, there is a button with a plus icon and the text 'Добавить перевод или переименование' (Add translation or rename).

Модуль позволяет задать свой вариант текстов в интерфейсе. Форма редактирования интерфейса имеет следующие поля:

The screenshot shows the 'Добавить новый' (Add new) form for adding a translation. It contains two text input fields: 'Исходное слово (фраза)' (Original word (phrase)) and 'Перевод (переименование)' (Translation (rename)). Below these fields is a dropdown menu labeled 'Для языка' (For language) with 'ru' selected and a question mark icon.

Перевод исходных текстов можно задавать вручную. Если открыта отличная от русской версия сайта, то весь доступный на текущей странице текст, не имеющий перевода, будет добавлен в базу для дальнейшего перевода. Искать добавленный текст легко с помощью формы поиска.

Перевод интерфейса можно импортировать и экспортировать, а также редактировать из пользовательской части с помощью модуля [«Панель быстрого редактирования»](#).

Подключение

Подключаемая часть – файл `modules/languages/languages.inc.php`. В нем описан класс `Languages_inc`. В модуле к объекту класса можно обратиться через переменную `$this`.

>diafan->_languages. Экземпляр класса создается при первом вызове переменной.

Методы

string **get (string \$name, [string \$module_name = "], [boolean \$useradmin=false], [array \$args = array()])** – возвращает перевод строки.

- access: public
- string \$name: текст для перевода
- string \$module_name: модуль
- boolean \$useradmin: выдавать форму для редактирования
- array \$args: значения, подставляемые в перевод строки, с помощью функции [vsprintf\(\)](#)

Для быстрого доступа к функции в основном объекте системы определена функция [_\(\)](#):

Для административной части

string **_ (string \$name, [boolean \$useradmin = false])** – возвращает перевод строки для текущего модуля.

- access: public
- string \$name: текст для перевода
- boolean \$useradmin: выдавать форму для редактирования, параметр только для пользовательской части

Дополнительно можно передать аргументы в функцию, для того чтобы подставить их в строку функцией [vsprintf\(\)](#).

Пример:

// полный вариант

```
$text = $this->diafan->_languages->get('Здравствуйте %s! У Вас %d новых сообщений.', 'registration', false, array('Иван Иванович', 3));
```

// сокращенный вариант

```
$text = $this->diafan->_('Здравствуйте %s! У Вас %d новых сообщений.', false, 'Иван Иванович', 3);
```

Шаблонные теги

Для работы с модулем «Языки сайта» служат следующие [шаблонные теги](#):

show_block – выводит ссылки на альтернативные языковые версии сайта.

Атрибуты:

— **template** – шаблон тега (файл *modules/languages/views/languages.view.show_block_template.php*; по умолчанию шаблон *modules/languages/views/languages.view.show_block.php*).

Пример:

```
<insert name="show_block" module="languages">
```

выведет список ссылок на языковые версии сайта, «ru/eng»

Файлы

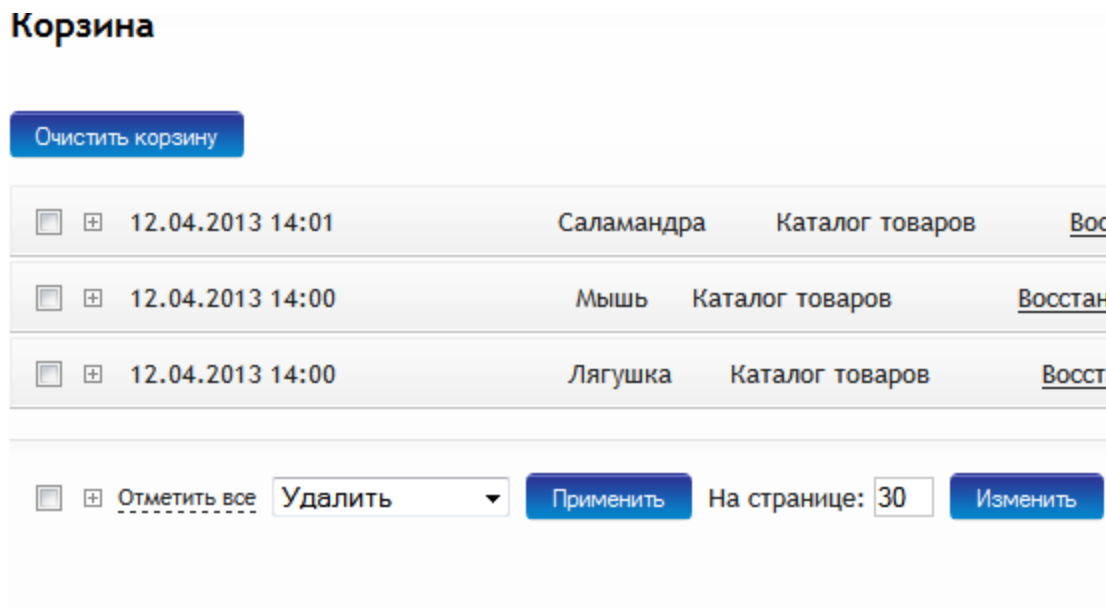
1. *modules/languages/admin/languages.admin.php* – редактирование списка языковых версий сайта;
2. *modules/languages/admin/languages.admin.js* – редактирование списка языковых версий сайта, JS-сценарий;
3. *modules/languages/admin/languages.admin.translate.php* – редактирование перевода интерфейса;
4. *modules/languages/languages.php* – контроллер;
5. *modules/languages/languages.export.php* – экспорт языкового файла;
6. *modules/languages/languages.inc.php* – подключение модуля;
7. *modules/languages/languages.model.php* – модель;
8. *modules/languages/views/languages.view.show_block.php* – шаблон ссылок на языковые версии сайта.

Корзина

- [Файлы](#)

Модуль относится к части CMS «Управление» и является обязательным для установки.

Все удаленные элементы на сайте попадают в корзину и их затем можно восстановить или окончательно удалить.



В модуль попадают сами элементы и принадлежащие им элементы, например, вместе с товаром удаляются комментарии, оценки, фотографии, псевдоссылки, дополнительные характеристики и т.д.

Модуль показывает имя пользователя, удалившего элементы.

Файлы

1. *modules/trash/admin/trash.admin.php* – корзина с удаленными элементами.

Модули

Модуль – часть сайта, работающая по определенному алгоритму. Модуль прикрепляется к одной или нескольким страницам сайта в административной части сайта и выполняет некоторые действия по формированию страницы в пользовательской части сайта.

Новости

Модуль новостей с поддержкой RSS-лент. Позволяет создавать новости в любом количестве групп и событий.

Магазин

Модуль «Магазин» позволяет решать большинство простых задач для управления интернет-торговлей с возможностью выгрузки в «Яндекс.Маркет». Пользовательские группы товаров, редактируемые характеристики и свойства товаров, зависимые характеристики, архив заказов со статусами, импорт и экспорт товаров и категорий, уведомления на e-mail и многое другое.

Фотогалерея

Удобное управление большим количеством фотографий. Можно создавать любое количество альбомов и фотографий.

Статьи

Публикация всевозможных статей на сайте с разделением на категории и подкатегории.

Баннеры

Вывод баннеров на сайте.

Файловый архив

Удобное управление большим количеством объектов, предназначенных для скачивания. Можно создавать любое количество категорий, иллюстраций к файлам и вложений.

Объявления

Универсальный модуль для создания базы данных объявлений, анкет, отзывов и пр. Элементы модуля можно группировать по категориям, добавлять через форму на сайте, задавать поля с помощью конструктора.

Теги

Облако тегов – представление списка ярлыков разным размером, в зависимости от частоты упоминаний. Теги можно подключать к любым объектам CMS от новостей до страниц сайта.

Поиск

Форма и результаты поиска по сайту.

Вопрос-Ответ

Управление вопросами пользователей сайта и ответами на них. Возможность группировать вопросы по категориям, отправка ответа на e-mail и функция «часто задаваемые вопросы» делают модуль очень полезным при общении с посетителями сайта.

Обратная связь

Сбор отзывов и пожеланий с сайта.

Форум

Общение пользователей на сайте.

Опросы

Благодаря модулю опросов владельцы сайтов могут собирать любую статистическую информацию с посетителей сайта.

Рассылки

Массовое информирование пользователей сайта о событиях сайта. Гибкая настройка групп подписчиков и тем рассылок.

Карта сайта

Создание страницы с картой всех ссылок сайта.

Ошибка на сайте

Получение сообщений от пользователей об ошибках в текстах сайта.

On-line консультант

Форма для связи пользователя с консультантом в режиме on-line.

Web Effector

Эффективный автоматический сервис продвижения сайтов в ТОП поисковых систем.

Файловый менеджер

Работа с файловой системой сайта – редактирование скриптов, загрузка и удаление файлов.

Новости

- [Новости](#)
- [Категории](#)
- [Настройки](#)
- [Шаблонные теги](#)
- [Файлы](#)

Модуль «Новости» отображает текстово-графические материалы с сортировкой по датам.

Модуль относится к части CMS «Контент» и может быть установлен опционально.

Модуль «Новости» можно прикреплять к различным страницам сайта. Если модуль прикреплен к нескольким страницам, в административной части модуля появляется специальный фильтр «Раздел сайта» с возможностью фильтрации вывода (списка новостей, категорий и настроек) по принадлежности к странице.

Модуль состоит из 2х частей: категории новостей и новости.

Новости

	Название	Дата	Статус
☐	О зверятах	11.04.2013	Птица-клён
			Продается ч... форме клен...
☐	О зверятах	10.04.2013	Мухомор...
			Намедни по...

В списке новости сортируются по дате.

Кнопка ✕ служит для удаления новости. Кнопка ⏻ служит для быстрой публикации новости (или, наоборот, временной деактивации новости).

Редактировать 

Название ?

Дата  ?

Показывать на сайте ☒ ?

Изображения 

Новость имеет следующие характеристики.

Основные:

- **Название новости** – Используется в ссылках на новость, заголовках. Параметр имеет разные значения для разных языковых версий сайта.
- **Дата** – Вводится в формате дд.мм.гггг чч:мм. Если указать будущую дату, новость начнет отображаться с этой даты.
- **Опубликовать на сайте** – Если не отмечена, новость не увидят посетители сайта. Параметр имеет разные значения для разных языковых версий сайта.
- **Важно (всегда сверху)** – Если отмечена, новость выведется в начале списка, независимо от сортировки по дате. Если важных новостей несколько, между собой они будут сортироваться по дате.
- **Изображения** – Изображения будут загружены автоматически после выбора. После загрузки изображения будут обработаны автоматически, согласно настройкам модуля. Параметр выводится, если в настройках модуля отмечена опция «Использовать изображения».
- **Похожие новости** – Выбор и добавление к текущей новости связей с другими новостями. Похожие новости выводятся шаблонным тегом `show_block_rel`. По умолчанию связи между новостями являются односторонними, это можно изменить, отметив опцию «В блоке похожих новостей связь двусторонняя» в настройках модуля.
- **Раздел сайта** – Перенос новости на другую страницу сайта, к которой прикреплен модуль новостей. Параметр выводится, если в настройках модуля отключена опция «Использовать категории», если опция подключена, то раздел сайта задается такой же, как у основной категории.
- **Категория** – Категория, к которой относится новость. Возможно выбрать дополнительные категории, в которых новость также будет выводиться. Чтобы выбрать несколько категорий, удерживайте CTRL. Параметр выводится, если в настройках модуля отмечена опция «Использовать категории».
- **Теги** – Добавление тегов к новости. Можно добавить либо новый тег, либо открыть и выбрать из уже существующих тегов. Параметр выводится, если в настройках модуля включен параметр «Подключить теги».
- **Счетчик просмотров** – Количество просмотров на сайте текущей новости. Статистика ведется и параметр выводится, если в настройках модуля отмечена опция «Подключить счетчик просмотров».

- **Комментарии** – Комментарии, которые оставили пользователи к текущей новости. Параметр выводится, если в настройках модуля включен параметр «Показывать комментарии к новостям».
- **Рейтинг** – Средний рейтинг, согласно голосованию пользователей сайта. Параметр выводится, если в настройках модуля включен параметр «Подключить рейтинг к новостям».
- **Анонс** – Краткое начало новости. Выводится в списках новостей и в блоках. Если отметить «Добавлять к описанию», на странице элемента анонс выведется вместе с основным описанием. Иначе анонс выведется только в списке, а на отдельной странице будет только описание. Если отметить «Применить типограф», контент будет отформатирован согласно правилам экранной типографики с помощью [веб-сервиса «Типограф»](#). Параметр имеет разные значения для разных языковых версий сайта.
- **Основной контент новости** – Полный текст для страницы новости. Если отметить «Применить типограф», контент будет отформатирован согласно правилам экранной типографики с помощью [веб-сервиса «Типограф»](#). Параметр имеет разные значения для разных языковых версий сайта.
- **Индексирование для поиска** – Новость автоматически индексируется для модуля «Поиск по сайту» при внесении изменений.
- **Индексирование для карты сайта** – Новость автоматически индексируется для карты сайта *sitemap.xml*.

Дополнительные:

- **Номер** – Номер элемента в БД (веб-мастеру и программисту).
- **Редактор** – Изменяется после первого сохранения. Показывает, кто из администраторов сайта первый правил текущую страницу.
- **Время последнего изменения** – Изменяется после сохранения элемента. Отдается в заголовке *Last Modify*.
- **Заголовок окна в браузере, тег Title** – Если не заполнен, тег *Title* будет автоматически сформирован как «Названия новости – Название страницы – Название сайта», либо согласно шаблонам автоформирования из настроек модуля (SEO-специалисту). Параметр имеет разные значения для разных языковых версий сайта.
- **Ключевые слова, тег Keywords** – Если не заполнен, тег *Keywords* будет автоматически сформирован согласно шаблонам автоформирования из настроек модуля (SEO-специалисту). Параметр имеет разные значения для разных языковых версий сайта.
- **Описание, тег Description** – Если не заполнен, тег *Description* будет автоматически сформирован согласно шаблонам автоформирования из настроек модуля (SEO-специалисту). Параметр имеет разные значения для разных языковых версий сайта.
- **Псевдоссылка** – ЧПУ, т.е. адрес страницы вида: <http://site.ru/pseudossylka/>. Смотрите параметры сайта (SEO-специалисту).
- **Редирект на текущую страницу со страницы** – Позволяет делать редирект с указанной страницы на текущую.
- **Changefreq** – Вероятная частота изменения этой страницы. Это значение используется для генерирования файла *sitemap.xml*. Подробнее читайте в описании [XML-формата файла Sitemap](#) (SEO-специалисту).
- **Priority** – Приоритетность URL относительно других URL на Вашем сайте. Это значение используется для генерирования файла *sitemap.xml*. Подробнее читайте в описании [XML-формата файла Sitemap](#) (SEO-специалисту).
- **Не показывать на карте сайта** – Скрывает отображение ссылки на новость в файле

sitemap.xml и [модуле «Карта сайта»](#).

- **Доступ к новости** – Если отметить опцию «Доступ только», новость увидят только авторизованные на сайте пользователи, отмеченных типов. Не авторизованные, в том числе поисковые роботы, увидят «404 Страница не найдена» (администратору сайта).
- **Период показа** – Если заполнить, текущая новость будет опубликована на сайте в указанный период. В иное время пользователи сайта новость не будут видеть, получая ошибку 404 «Страница не найдена» (администратору сайта).
- **Шаблон страницы** – Возможность подключить для страницы новости шаблон сайта отличный от основного (*themes/site.php*). Все шаблоны для сайта должны храниться в папке *themes* с расширением *.php* (например, *themes/dizain_so_slajdom.php*). Подробнее в [разделе «Шаблоны сайта»](#). (веб-мастеру и программисту, не меняйте этот параметр, если не уверены в результате!).
- **Шаблон модуля** – Шаблон вывода контента модуля на странице отдельной новости (веб-мастеру и программисту, не меняйте этот параметр, если не уверены в результате!).

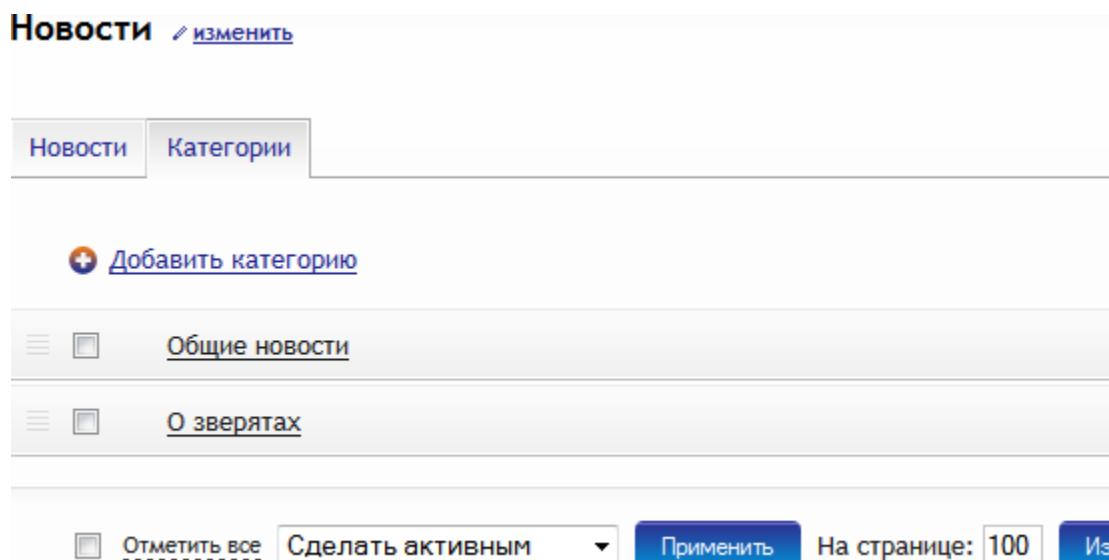
В модуль «Новости» встроена автоматическая публикация новостей в формате RSS.

Адрес RSS-ленты: <http://site.ru/news/rss/>.

У англоязычной версии сайта адрес RSS-ленты новостей будет иметь вид:
<http://site.ru/eng/news/rss/>.

Категории

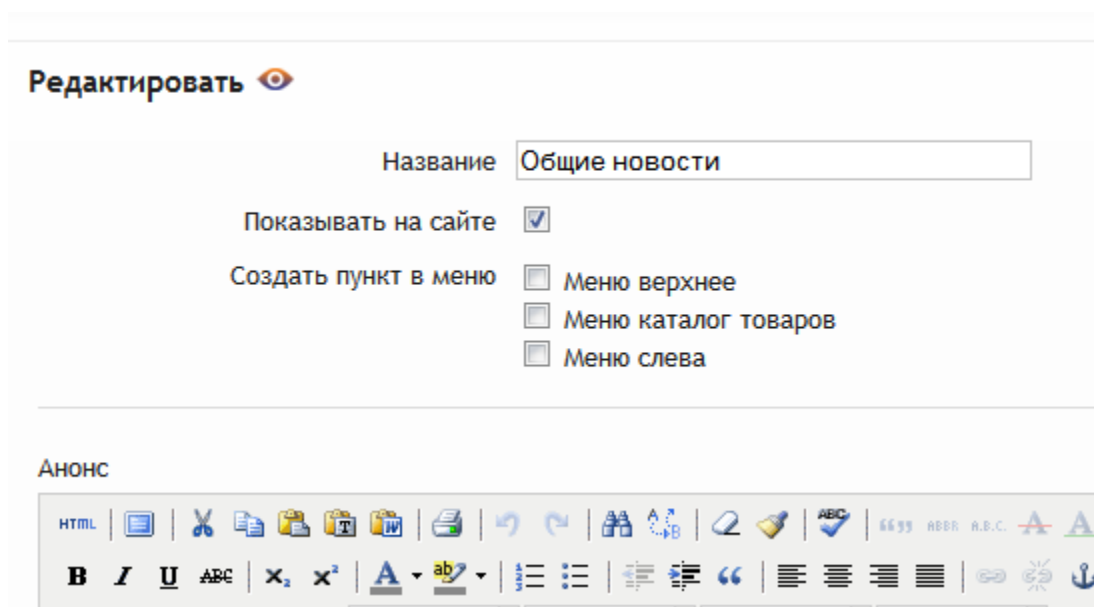
Категории новостей могут иметь неограниченное число подкатегорий. При необходимости категории можно не использовать. Для этого нужно отключить параметр «Использовать категории» в настройках модуля.



Если перед названием категории имеется знак +, то этой категории принадлежат подкатегории. Добавить подкатеорию к категории можно нажав на знак  рядом с названием категории.

Порядок отображения категорий можно менять с помощью мыши простым перетаскиванием.

Кнопка ✕ служит для удаления категории. Кнопка ⏻ служит для быстрой публикации категории (или, наоборот, временного скрытия на сайте). При удалении категории удаляются все вложенные подкатегории и новости.



Редактировать 

Название

Показывать на сайте ☒

Создать пункт в меню ☐ Меню верхнее
☐ Меню каталог товаров
☐ Меню слева

Анонс

HTML |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |

- **Индексирование для карты сайта** – Категория автоматически индексируется для карты сайта *sitemap.xml*.

Дополнительные:

- **Номер** – Номер элемента в БД (веб-мастеру и программисту).
- **Редактор** – Изменяется после первого сохранения. Показывает, кто из администраторов сайта первый правил текущую страницу.
- **Время последнего изменения** – Изменяется после сохранения элемента. Отдается в заголовке *Last Modify*.
- **Заголовок окна в браузере, тег Title** – Если не заполнен, тег *Title* будет автоматически сформирован как «Название категории – Название страницы – Название сайта», либо согласно шаблонам автоформирования из настроек модуля (SEO-специалисту). Параметр имеет разные значения для разных языковых версий сайта.
- **Ключевые слова, тег Keywords** – Если не заполнен, тег *Keywords* будет автоматически сформирован согласно шаблонам автоформирования из настроек модуля (SEO-специалисту). Параметр имеет разные значения для разных языковых версий сайта.
- **Описание, тег Description** – Если не заполнен, тег *Description* будет автоматически сформирован согласно шаблонам автоформирования из настроек модуля (SEO-специалисту). Параметр имеет разные значения для разных языковых версий сайта.
- **Псевдоссылка** – ЧПУ, т.е. адрес страницы вида: <http://site.ru/pseudossylka/>. Смотрите параметры сайта (SEO-специалисту).
- **Редирект на текущую страницу со страницы** – Позволяет делать редирект с указанной страницы на текущую.
- **Changefreq** – Вероятная частота изменения этой страницы. Это значение используется для генерирования файла *sitemap.xml*. Подробнее читайте в описании [XML-формата файла Sitemap](#) (SEO-специалисту).
- **Priority** – Приоритетность URL относительно других URL на Вашем сайте. Это значение используется для генерирования файла *sitemap.xml*. Подробнее читайте в описании [XML-формата файла Sitemap](#) (SEO-специалисту).
- **Не показывать на карте сайта** – Скрывает отображение ссылки на категорию в файле *sitemap.xml* и [модуле «Карта сайта»](#).
- **Раздел сайта** – Перенос категории на другую страницу сайта, к которой прикреплен модуль (администратору сайта).
- **Вложенность: принадлежит** – Перемещение текущей категории и всех её подкатегорий в принадлежность другой категории (администратору сайта).
- **Сортировка: установить перед** – Редактирование порядка следования категории в списке. Поле доступно для редактирования только для категорий, отображаемых на сайте.
- **Доступ** – Если отметить опцию «Доступ только», категорию увидят только авторизованные на сайте пользователи, отмеченных типов. Не авторизованные, в том числе поисковые роботы, увидят «404 Страница не найдена» (администратору сайта).
- **Шаблон страницы** – Возможность подключить для страницы категории шаблон сайта отличный от основного (*themes/site.php*). Все шаблоны для сайта должны храниться в папке *themes* с расширением *.php* (например, *themes/dizain_so_slajdom.php*). Подробнее в [разделе «Шаблоны сайта»](#). (веб-мастеру и программисту, не меняйте этот параметр, если не уверены в результате!).
- **Шаблон модуля** – Шаблон вывода контента модуля на странице списка новостей в категории (веб-мастеру и программисту, не меняйте этот параметр, если не уверены в

результате!).

- **Шаблон модуля для вложенных новостей** – Шаблон вывода контента модуля на странице отдельной новости, вложенной в текущую категорию (веб-мастеру и программисту, не меняйте этот параметр, если не уверены в результате!).

Настройки

Можно сохранять разные настройки модуля для разных страниц, к которым прикреплен модуль.

Настройки

Количество новостей на странице

Формат даты

Использовать категории ☒

Количество новостей в списке категорий

Количество новостей в списке вложенной категории

Показывать новости подкатегорий ☐

- **Количество новостей на странице** – Количество одновременно выводимых новостей в списке.
- **Формат даты** – Позволяет настроить отображение даты в модуле. Возможные значения:
 - 01.05.2013,
 - 01.05.2013 14:45,
 - 1 мая 2013 г.,
 - 1 мая,
 - 1 мая 2013, понедельник,
 - вчера 15:30,
 - не отображать.
- **Использовать категории** – Разделение новостей на категории, рубрики.
- **Количество новостей в списке категорий** – Количество новостей, выводимых в списке категорий на главной странице модуля. Параметр появляется, если отмечена опция «Использовать категории».
- **Количество новостей в списке вложенной категории** – Для первой страницы модуля и для страницы категории. Параметр появляется, если отмечена опция «Использовать категории».
- **Показывать новости подкатегорий** – Если отмечена, в списке новостей категории будут отображаться последние новости из всех вложенных категорий. Параметр появляется, если отмечена опция «Использовать категории».
- **Использовать изображения** – Если отмечена, к новостям можно будет добавлять изображения.

- **Генерировать размеры изображений** – Размеры изображений, заданные в модуле «Изображения» и тег латинскими буквами для подключения изображения на сайте. Обязательно должны быть заданы два размера: превью изображения в списке новостей (тег medium) и полное изображение (тег large).
- **Отображение изображений в списке** – Параметр принимает значения:
 - нет (отключает отображение изображений в списке);
 - показывать одно изображение;
 - показывать все изображения. Параметр выводится, если отмечена опция «Использовать изображения».
- **Использовать изображения для категорий** – Позволяет включить/отключить загрузку изображений к категориям.
- **Генерировать размеры изображений для категорий** – Размеры изображений, заданные в модуле «Изображения» и тег латинскими буквами для подключения изображения на сайте. Обязательно должны быть заданы два размера: превью изображения в списке категорий (тег medium) и полное изображение (тег large). Параметр выводится, если отмечена опция «Использовать изображения для категорий».
- **Отображение изображений в списке категорий** – Параметр принимает значения:
 - нет (отключает отображение изображений в списке);
 - показывать одно изображение;
 - показывать все изображения. Параметр выводится, если отмечена опция «Использовать изображения для категорий».
- **Использовать анимацию при увеличении изображений** – Параметр добавляет JavaScript код, позволяющий включить анимацию при увеличении изображений. Параметр выводится, если отмечена опция «Использовать изображения».
- **Максимальный размер загружаемых файлов** – Параметр показывает максимально допустимый размер загружаемых файлов, установленный в настройках хостинга. Параметр выводится, если отмечена опция «Использовать изображения».
- **Применить настройки ко всем ранее загруженным изображениям** – Позволяет перекодировать размер уже загруженных изображений. Кнопка необходима, если изменены настройки размеров изображений. Параметр выводится, если отмечена опция «Использовать изображения».
- **Подключить счетчик просмотров** – Позволяет считать количество просмотров отдельной новости.
- **Выводить счетчик на сайте** – Позволяет вывести на сайте количество просмотров отдельной новости. Параметр выводится, если отмечена опция «Счетчик просмотров». Параметр появляется, если отмечена опция «Подключить счетчик просмотров».
- **Подключить комментарии к новостям** – Если отмечена, пользователи сайта смогут комментировать новости. Параметр не будет включен, если модуль «Комментарии» не установлен. Подробности см. в разделе [модуль Комментарии](#).
- **Показывать комментарии к категориям** – Подключение модуля «Комментарии» к категориям новостей. Параметр не будет включен, если модуль «Комментарии» не установлен. Подробности см. в разделе [модуль Комментарии](#).
- **Подключить теги** – Если отмечена, к каждой новости можно будет добавлять теги. Параметр не будет включен, если модуль «Теги» не установлен. Подробности см. в разделе [модуль «Теги»](#).
- **Подключить рейтинг новостей** – Если отмечена, каждой новости пользователи сайта смогут выставить рейтинг. Параметр не будет включен, если модуль «Рейтинг» не установлен. Подробности см. в разделе [модуль «Рейтинг»](#).
- **Подключить рейтинг к категориям** – Подключение модуля «Рейтинг» к категориям.

Параметр не будет включен, если модуль «Рейтинг» не установлен. Подробности см. в разделе [модуль «Рейтинг»](#).

- **Подключить перелинковку** – Отображение перелинковки в модуле. Подробности см. в разделе [модуль «Перелинковка»](#).
- **В блоке похожих новостей связь двусторонняя** – Если отметить, то при назначении новости А похожей новости Б, у новости Б автоматически станет похожая новость А.
- **Шаблон для автоматического генерирования Title** – Если шаблон задан, то заголовок автоматически генерируется по шаблону. В шаблон можно добавить:
 - %title – заданный заголовок,
 - %name – название,
 - %category – название категории,
 - %parent_category – название категории верхнего уровня (SEO-специалисту).

Параметр имеет разные значения для разных языковых версий сайта.

- **Шаблон для автоматического генерирования Title для категории** – Если шаблон задан, то заголовок автоматически генерируется по шаблону. В шаблон можно добавить:
 - %title – заданный заголовок,
 - %name – название категории,
 - %parent – название категории верхнего уровня,
 - %page – страница (текст можно поменять в интерфейсе «Языки сайта» – «Перевод интерфейса») (SEO-специалисту).

Параметр имеет разные значения для разных языковых версий сайта. Параметр появляется, если отмечена опция «Использовать категории».

- **Шаблон для автоматического генерирования Keywords** – Если шаблон задан, то поле *Keywords* автоматически генерируется по шаблону. В шаблон можно добавить:
 - %keywords – заданные ключевые слова,
 - %name – название,
 - %category – название категории,
 - %parent_category – название категории верхнего уровня (SEO-специалисту).

Параметр имеет разные значения для разных языковых версий сайта.

- **Шаблон для автоматического генерирования Keywords для категории** – Если шаблон задан, то поле *Keywords* автоматически генерируется по шаблону. В шаблон можно добавить:
 - %keywords – заданные ключевые слова,
 - %name – название категории,
 - %parent – название категории верхнего уровня (SEO-специалисту).

Параметр имеет разные значения для разных языковых версий сайта. Параметр появляется, если отмечена опция «Использовать категории».

- **Шаблон для автоматического генерирования Description** – Если шаблон задан, то поле *Description* автоматически генерируется по шаблону. В шаблон можно добавить:
 - %descr – заданное описание,
 - %name – название,
 - %category – название категории,
 - %parent_category – название категории верхнего уровня (SEO-специалисту).

Параметр имеет разные значения для разных языковых версий сайта.

- **Шаблон для автоматического генерирования Description для категории** – Если шаблон задан, то поле *Description* автоматически генерируется по шаблону. В шаблон

можно добавить %descr - заданное описание, %name - название категории, %parent - название категории верхнего уровня (SEO-специалисту). Параметр имеет разные значения для разных языковых версий сайта. Параметр появляется, если отмечена опция «Использовать категории».

- **Шаблон для списка элементов** – По умолчанию *modules/news/views/news.view.list.php*. Параметр для разработчиков! Не устанавливайте, если не уверены в результате.
- **Шаблон для первой страницы модуля (если подключены категории)** – По умолчанию *modules/news/views/news.view.first_page.php*. Параметр для разработчиков! Не устанавливайте, если не уверены в результате.
- **Шаблон для страницы новости** – По умолчанию, *modules/news/views/news.view.id.php*. Параметр для разработчиков! Не устанавливайте, если не уверены в результате.
- **Отдельный пункт в меню администрирования для каждого раздела сайта** – Если модуль подключен к нескольким страницам сайта, отметка данного параметра выведет несколько пунктов в меню административной части для удобства быстрого доступа (администратору сайта).
- **Индексирование для карты сайта** – При изменении настроек, влияющих на отображение страницы, модуль автоматически переиндексируется для карты сайта *sitemap.xml*.

Шаблонные теги

Для работы с модулем «Новости» служат следующие [шаблонные теги](#):

show_block – выводит последние новости на всех страницах, кроме страницы новостей, когда выводится список тех же новостей, что и в функции.

Атрибуты:

— **count** – количество выводимых новостей (по умолчанию 3);

— **site_id** – страницы, к которым прикреплен модуль. Идентификаторы страниц перечисляются через запятую. По умолчанию выбираются все страницы;

— **cat_id** – категории новостей, если в настройках модуля отмечено «Использовать категории»;

— **sort** – сортировка новостей: **date** – по дате (по умолчанию), **keywords** – новости, похожие по названию для текущей страницы;

— **images** – количество изображений, прикрепленных к новости;

— **images_variation** – тег размера изображений, задается в настройках модуля;

— **template** – шаблон тега (файл *modules/news/views/news.view.show_block_template.php*; по умолчанию шаблон *modules/news/views/news.view.show_block.php*).

Формат даты задается в настройках модуля.

Пример:

```
<insert name="show_block" module="news">
```

выведет 3 последние новости сайта

```
<insert name="show_block" module="news" count="5">
```

выведет 5 последних новостей сайта

```
<insert name="show_block" module="news" count="1" site_id="12">
```

выведет последнюю новость, прикрепленную к странице 12

show_block_rel – на странице новости выводит похожие новости. По умолчанию связи между новостями являются односторонними, это можно изменить, отметив опцию «В блоке похожих новостей связь двусторонняя» в настройках модуля.

Атрибуты:

- **count** – количество выводимых новостей (по умолчанию 3);
- **images** – количество изображений, прикрепленных к новости;
- **images_variation** – тег размера изображений, задается в настройках модуля;
- **template** – шаблон тега (файл *modules/news/views/news.view.show_block_rel_***template***.php*; по умолчанию шаблон *modules/news/views/news.view.show_block_rel.php*).

Формат даты задается в настройках модуля.

Пример:

```
<insert name="show_block_rel" module="news">
```

выведет 3 новости, прикрепленные к текущей новости

show_calendar – выводит календарь со ссылками на новости за период. Периоды отображаются в виде ссылок на месяцы, только если имеются новости, соответствующие этим периодам.

Атрибуты:

- **site_id** – страница, к которой прикреплен модуль, по умолчанию выбирается одна страница;
- **cat_id** – категория новостей (id категории, по умолчанию учитываются все новости),

если в настройках модуля отмечено «Использовать категории»;

— **only_news** – выводить блок только на странице «Новости»;

— **detail** – детализация (**day** – дни, **month** – месяца (по умолчанию), **year** – годы);

— **template** – шаблон тега (файл

*modules/news/views/news.view.show_calendar_***template***.php* или

*modules/news/views/news.view.show_calendar_day_***template***.php* для детализации по дням; по умолчанию шаблон *modules/news/views/news.view.show_calendar.php* или *modules/news/views/news.view.show_calendar_day.php* для детализации по дням).

Пример:

```
<insert name="show_calendar" module="news">
```

выведет календарь новостей

Файлы

Модуль находится в папке *modules/news* и устроен [классически](#)..

1. *modules/news/admin/news.admin.php* – редактирование новостей;
2. *modules/news/admin/news.admin.category.php* – редактирование категорий новостей;
3. *modules/news/admin/news.admin.config.php* – настройки модуля;
4. *modules/news/admin/news.admin.counter.php* – статистика просмотров;
5. *modules/news/admin/news.admin.menu.php* – карта ссылок для модуля «Меню на сайте»;
6. *modules/news/news.php* – контроллер;
7. *modules/news/news.ajax.php* – обработка Ajax-запросов;
8. *modules/news/news.install.php* – установка модуля;
9. *modules/news/news.js* – JS-сценарий модуля;
10. *modules/news/news.model.php* – модель;
11. *modules/news/news.rss.php* – RSS лента новостей;
12. *modules/news/news.search.php* – настройки для поисковой индексации для модуля «Поиск»;
13. *modules/news/news.sitemap.php* – карта ссылок для модуля «Карта сайта»;
14. *modules/news/views/m/news.view.show_block.php* – шаблон блока новостей для мобильной версии;

15. *modules/news/views/news.view.first_page.php* – шаблон первой страницы модуля, если в настройках модуля подключен параметр «Использовать категории»;
16. *modules/news/views/news.view.id.php* – шаблон страницы новости;
17. *modules/news/views/news.view.list.php* – шаблон список новостей;
18. *modules/news/views/news.view.show_block.php* – шаблон блока новостей;
19. *modules/news/views/news.view.show_block_rel.php* – шаблон блока похожих новостей;
20. *modules/news/views/news.view.show_calendar.php* – шаблон календаря архива новостей;
21. *modules/news/views/news.view.show_calendar_day.php* – шаблон календаря архива новостей по дням.

Магазин

- [Товары](#)
- [Категории](#)
- [Характеристики](#)
- [Заказы](#)
- [Форма оформления заказа](#)
- [Отчеты](#)
- [Список пожеланий](#)
- [Список ожиданий](#)
- [Статусы заказа](#)
- [Скидки](#)
- [Валюты](#)
- [Методы оплаты](#)
- [Способы доставки](#)
- [Сопутствующие услуги](#)
- [История платежей](#)
- [Импорт/экспорт \(+Я.Маркет\)](#)
- [Интеграция с системой 1С:Предприятие](#)
- [Интеграция с системой МойСклад](#)
- [Настройки модуля](#)
- [Шаблонные теги](#)
- [Файлы](#)

Модуль относится к части CMS «Интернет магазин» и может быть установлен опционально.

Модуль «Магазин» можно прикреплять к различным страницам сайта. Если модуль прикреплен к нескольким страницам, в административной части модуля появляется специальный фильтр «Раздел сайта» с возможностью фильтрации вывода (списка товаров, категорий и настроек) по принадлежности к странице.

Товары

Каталог товаров [изменить](#)

Товары	Характеристики	Категории	Импорт/экспорт
--------	----------------	-----------	----------------

Категория: Все

Название:

Товар временно отсутствует: ☐

Найти

[Добавить товар](#)

Птицы	Коршун	3100 руб.↓	Отличается: крылья остр
Рыбка-	100	руб.	Тело высок

Порядок отображения товаров можно редактировать с помощью мыши и кнопки . Если подключены категории товаров, то сортировать товары можно только в пределах категории (в фильтре «Категория» нужно выбрать категорию товаров). Кнопка служит для удаления товаров. Кнопка служит для быстрой публикации товаров (или, наоборот, блокирования).

Редактировать

Название ?

Артикул

Показывать на сайте ☒ ?

Товар временно отсутствует ☐ ?

Хит ☒

Новинка ☐

Акция ☐

Товары имеют следующие характеристики.

Основные:

- **Название** – Используется в заголовках, ссылках на товар, при заказе. Параметр имеет разные значения для разных языковых версий сайта.
- **Цена** – Можно задать несколько вариантов цены и количества для товара. Для этого следует создать дополнительную характеристику с типом «список с выбором нескольких значений», отметить опцию «Доступен к выбору при заказе» и при редактировании товара отметить возле характеристики опцию «Влияет на цену». К

варианту товара можно прикрепить изображение из тех, что уже загружены для товара. Поле «Количество» отображается, если в настройках модуля подключена опция «Учитывать остатки товаров на складе». Если поле «Количество» пустое и в настройках не отмечена опция «Разрешать покупать товары без цены», то товар нельзя купить.

- **Артикул** – Внутренний артикул товара. Если заполнить, будет выводиться на сайте и использоваться в поиске по товарам.
- **Раздел сайта** – Перенос товара на другую страницу сайта, к которой прикреплен модуль. Параметр выводится, если в настройках модуля отключена опция «Использовать категории», если опция подключена, то раздел сайта задается такой же, как у основной категории.
- **Категория** – Категория, к которой относится товар. Возможно выбрать дополнительные категории, в которых товар также будет выводиться. Чтобы выбрать несколько категорий, удерживайте CTRL. Параметр выводится, если в настройках модуля отмечена опция «Использовать категории».
- **Опубликовать на сайте** – Если не отмечена, страница товара не будет выводиться на сайте. Параметр имеет разные значения для разных языковых версий сайта.
- **Товар временно отсутствует** – Если отмечена, у товара не будет кнопки «Купить», выведется сообщение «Товар временно отсутствует», а посетители сайта смогут подписаться на уведомление о поступлении. Подписанных посетителей можно будет увидеть в разделе «Статистика» - «Список ожиданий». В случае поступления товара и снятия данной галки, все подписанные посетители автоматически получают уведомления о поступлении товара на указанный электронный ящик.
- **Хит** – Если отмечена, товар будет участвовать в поиске по соответствующему признаку, в списках и карточке товара будет выводиться соответствующий значёк. Товары можно будет группировать.
- **Новинка** – Если отмечена, товар будет участвовать в поиске по соответствующему признаку, в списках и карточке товара будет выводиться соответствующий значёк. Товары можно будет группировать.
- **Акция** – Если отмечена, товар будет участвовать в поиске по соответствующему признаку, в списках и карточке товара будет выводиться соответствующий значёк. Товары можно будет группировать.
- **Загрузить файл** – Цифровой товар. Загрузите файл, который необходимо продавать. Он будет храниться в закрытой папке без прямого доступа, а ссылка на его скачивание будет формироваться после оплаты. Параметр выводится, если в настройках модуля отмечена опция «Включить возможность продажи файлов».
- **Изображения** – Иллюстрации к товару. Можно загрузить сразу несколько. Иллюстрации загрузятся автоматически после выбора. Варианты размера загружаемых изображений определяются в настройках. Параметр выводится, если в настройках модуля отмечена опция «Использовать изображения».
- **Скидки** – Список всех доступных скидок. Возможность подключить или отключить скидку для товара.
- **Характеристики** – Группа полей, определенных в части «Характеристики». Для характеристики с типом «список с выбором нескольких значений» можно задать несколько цен для товара. Для одного товара можно выбрать несколько характеристик, влияющих на цену. Параметр имеет разные значения для разных языковых версий сайта.
- **Похожие товары** – Выбор и добавление к текущему товару связей с другими товарами. Похожие товары выводятся шаблонным тегом `show_block_rel`. По умолчанию связи между товарами являются односторонними, это можно изменить, отметив опцию «В блоке похожих товаров связь двусторонняя» в настройках модуля.
- **Теги** – Добавление тегов к товару. Можно добавить либо новый тег, либо открыть и

выбрать из уже существующих тегов. Параметр выводится, если в настройках модуля включен параметр «Подключить теги».

- **Статистика** – Счетчик просмотров и покупок текущего товара. Статистика просмотров ведется и параметр «Просмотров товара» выводится, если в настройках модуля отмечена опция «Подключить счетчик просмотров».
- **Комментарии** – Комментарии, которые оставили пользователи к текущему товару. Параметр выводится, если в настройках модуля включен параметр «Показывать комментарии к товарам».
- **Рейтинг** – Средний рейтинг, согласно голосованию пользователей сайта. Параметр выводится, если в настройках модуля включен параметр «Подключить рейтинг к товарам».
- **Анонс** – Краткое описание товара. Выводится в списках товара и в блоках. Если отметить «Добавлять к описанию», на странице товара анонс выведется вместе с основным описанием. Иначе анонс выведется только в списке, а на отдельной странице будет только описание. Если отметить «Применить типограф», контент будет отформатирован согласно правилам экранной типографики с помощью [веб-сервиса «Типограф»](#). Параметр имеет разные значения для разных языковых версий сайта.
- **Описание** – Полное описание для страницы товара. Если отметить «Применить типограф», контент будет отформатирован согласно правилам экранной типографики с помощью [веб-сервиса «Типограф»](#). Параметр имеет разные значения для разных языковых версий сайта.
- **Клонирование товара.**
- **Индексирование для поиска** – Товар автоматически индексируется для модуля «Поиск по сайту» при внесении изменений.
- **Индексирование для карты сайта** – Товар автоматически индексируется для карты сайта *sitemap.xml*.

Дополнительные:

- **Номер** – Номер элемента в БД (веб-мастеру и программисту).
- **Редактор** – Изменяется после первого сохранения. Показывает, кто из администраторов сайта первый правил текущую страницу.
- **Время последнего изменения** – Изменяется после сохранения элемента. Отдается в заголовке *Last Modify*.
- **Заголовок окна в браузере, тег Title** – Если не заполнен, тег *Title* будет автоматически сформирован как «Название товара – Название страницы – Название сайта», либо согласно шаблонам автоформирования из настроек модуля (SEO-специалисту). Параметр имеет разные значения для разных языковых версий сайта.
- **Ключевые слова, тег Keywords** – Если не заполнен, тег *Keywords* будет автоматически сформирован согласно шаблонам автоформирования из настроек модуля (SEO-специалисту). Параметр имеет разные значения для разных языковых версий сайта.
- **Описание, тег Description** – Если не заполнен, тег *Description* будет автоматически сформирован согласно шаблонам автоформирования из настроек модуля (SEO-специалисту). Параметр имеет разные значения для разных языковых версий сайта.
- **Псевдоссылка** – ЧПУ, т.е. адрес страницы вида: <http://site.ru/pseudossylka/>. Смотрите параметры сайта (SEO-специалисту).
- **Редирект на текущую страницу со страницы** – Позволяет делать редирект с указанной страницы на текущую.
- **Changefreq** – Вероятная частота изменения этой страницы. Это значение

используется для генерирования файла *sitemap.xml*. Подробнее читайте в описании [XML-формата файла Sitemap](#) (SEO-специалисту).

- **Priority** – Приоритетность URL относительно других URL на Вашем сайте. Это значение используется для генерирования файла *sitemap.xml*. Подробнее читайте в описании [XML-формата файла Sitemap](#) (SEO-специалисту).
- **Сортировка: установить перед** – Изменить положение текущего товара среди других товаров. Поле доступно для редактирования только для товаров, отображаемых на сайте (администратору сайта).
- **Период показа** – Если заполнить, текущий товар будет опубликован на сайте в указанный период. В иное время пользователи сайта товар не будут видеть, получая ошибку 404 «Страница не найдена» (администратору сайта).
- **Доступ** – Если отметить опцию «Доступ только», товар увидят только авторизованные на сайте пользователи, отмеченных типов. Не авторизованные, в том числе поисковые роботы, увидят «404 Страница не найдена» (администратору сайта).
- **Не показывать на карте сайта** – Скрывает отображение ссылки на товар в файле *sitemap.xml* и [модуле «Карта сайта»](#).
- **Выгружать в Яндекс.Маркет** – Параметр разрешит или запретит выгружать этот товар. Параметр выводится, если в настройках модуля отмечена опция «Подключить Яндекс Маркет» и параметр «Выгружать товары в Яндекс.Маркет» определен как «только помеченные».
- **Поля для Яндекс Маркет** – Параметры, необходимые для формирования информации для системы «Яндекс Маркет» файлом *modules/shop/shop.yandex.php*. Параметр выводится, если в настройках модуля отмечена опция «Подключить Яндекс Маркет». Параметр появляется, если отмечена опция «Выгружать в Яндекс.Маркет».
- **Идентификатор для импорта** – Можно заполнить для идентификации категории при импорте (администратору сайта).
- **Шаблон страницы** – Возможность подключить для страницы товара шаблон сайта отличный от основного (*themes/site.php*). Все шаблоны для сайта должны храниться в папке *themes* с расширением *.php* (например, *themes/dizain_so_slajdom.php*). Подробнее в [разделе «Шаблоны сайта»](#). (веб-мастеру и программисту, не меняйте этот параметр, если не уверены в результате!).
- **Шаблон модуля** – Шаблон вывода контента модуля на странице отдельного товара (веб-мастеру и программисту, не меняйте этот параметр, если не уверены в результате!).

Категории

Категории товаров могут иметь неограниченное число подкатегорий. При необходимости категории можно не использовать. Для этого нужно отключить параметр «Использовать категории» в настройках модуля.

Каталог товаров [изменить](#)

Товары	Характеристики	Категории	Импорт/экспорт
--------	----------------	-----------	----------------

 [Добавить категорию](#)

		Птицы (м)
	 	Звери (м)
		Водоплавающие (м)

Если перед названием категории имеется знак +, то этой категории принадлежат подкатегории. Добавить подкатеорию к категории можно нажав на знак  рядом с названием категории.

Порядок отображения категорий можно редактировать мышью с помощью кнопки .

Кнопка  служит для удаления категории. Кнопка  служит для быстрой публикации категории (или, наоборот, блокирования). При удалении категории удаляются все вложенные подкатегории и товары.

Редактировать 

Название

Показывать на сайте

☒

Создать пункт в меню

☐ Меню верхнее

☒ Меню каталог товаров

☐ Меню слева

Дополнительные параметры 

Номер id=1 ?

Заголовок окна в браузере, тэг Title

Категории имеют следующие характеристики.

Основные:

- **Название категории** – Используется в ссылках на категорию, заголовках. Параметр имеет разные значения для разных языковых версий сайта.
- **Опубликовать на сайте** – Если не отмечена, категорию не увидят посетители сайта.

Параметр имеет разные значения для разных языковых версий сайта.

- **Создать пункт в меню** – Если отметить, в [модуле «Меню на сайте»](#) будет создан пункт со ссылкой на текущую категорию.
- **Изображения** – Возможность загрузки нескольких изображений. Варианты размера загружаемых изображений определяются в настройках. Параметр выводится, если в настройках модуля отмечена опция «Использовать изображения для категорий».
- **Характеристики товаров** – Список характеристик, применимых к товарам текущей категории.
- **Комментарии** – Комментарии, которые оставили пользователи к текущей категории. Параметр выводится, если в настройках модуля включен параметр «Показывать комментарии к категориям».
- **Рейтинг** – Средний рейтинг, согласно голосованию пользователей сайта. Параметр выводится, если в настройках модуля включен параметр «Подключить рейтинг к категориям».
- **Краткое описание** – Краткое описание категории. Если отметить «Добавлять к описанию», на странице элемента анонс выведется вместе с основным описанием. Иначе анонс выведется только в списке, а на отдельной странице будет только описание. Если отметить «Применить типограф», контент будет отформатирован согласно правилам экранной типографики с помощью [веб-сервиса «Типограф»](#). Параметр имеет разные значения для разных языковых версий сайта.
- **Полное описание** – Если отметить «Применить типограф», контент будет отформатирован согласно правилам экранной типографики с помощью [веб-сервиса «Типограф»](#). Параметр имеет разные значения для разных языковых версий сайта.
- **Индексирование для поиска** – Категория автоматически индексируется для модуля «Поиск по сайту» при внесении изменений.
- **Индексирование для карты сайта** – Категория автоматически индексируется для карты сайта *sitemap.xml*.

Дополнительные:

- **Номер** – Номер элемента в БД (веб-мастеру и программисту).
- **Редактор** – Изменяется после первого сохранения. Показывает, кто из администраторов сайта первый правил текущую страницу.
- **Время последнего изменения** – Изменяется после сохранения элемента. Отдается в заголовке *Last Modify*.
- **Заголовок окна в браузере, тег Title** – Если не заполнен, тег *Title* будет автоматически сформирован как «Название категории – Название страницы – Название сайта», либо согласно шаблонам автоформирования из настроек модуля (SEO-специалисту). Параметр имеет разные значения для разных языковых версий сайта.
- **Ключевые слова, тег Keywords** – Если не заполнен, тег *Keywords* будет автоматически сформирован согласно шаблонам автоформирования из настроек модуля (SEO-специалисту). Параметр имеет разные значения для разных языковых версий сайта.
- **Описание, тег Description** – Если не заполнен, тег *Description* будет автоматически сформирован согласно шаблонам автоформирования из настроек модуля (SEO-специалисту). Параметр имеет разные значения для разных языковых версий сайта.
- **Псевдоссылка** – ЧПУ, т.е. адрес страницы вида: <http://site.ru/pseudossylka/>. Смотрите параметры сайта (SEO-специалисту).
- **Редирект на текущую страницу со страницы** – Позволяет делать редирект с указанной страницы на текущую.

- **Changefreq** – Вероятная частота изменения этой страницы. Это значение используется для генерирования файла *sitemap.xml*. Подробнее читайте в описании [XML-формата файла Sitemap](#) (SEO-специалисту).
- **Priority** – Приоритетность URL относительно других URL на Вашем сайте. Это значение используется для генерирования файла *sitemap.xml*. Подробнее читайте в описании [XML-формата файла Sitemap](#) (SEO-специалисту).
- **Не показывать на карте сайта** – Скрывает отображение ссылки на категорию в файле *sitemap.xml* и [модуле «Карта сайта»](#).
- **Выгружать в Яндекс Маркет** – Параметр разрешит или запретит выгружать эту категорию. Параметр выводится, если в настройках модуля отмечена опция «Подключить Яндекс Маркет» и параметр «Выгружать категории в Яндекс.Маркет» определен как «только помеченные».
- **Раздел сайта** – Перенос категории на другую страницу сайта, к которой прикреплен модуль (администратору сайта).
- **Вложенность: принадлежит** – Перемещение текущей категории и всех её подкатегорий в принадлежность другой категории (администратору сайта).
- **Сортировка: установить перед** – Редактирование порядка следования категории в списке. Поле доступно для редактирования только для категорий, отображаемых на сайте.
- **Доступ** – Если отметить опцию «Доступ только», категорию увидят только авторизованные на сайте пользователи, отмеченных типов. Не авторизованные, в том числе поисковые роботы, увидят «404 Страница не найдена» (администратору сайта).
- **Идентификатор для импорта** – Можно заполнить для идентификации категории при импорте (администратору сайта).
- **Шаблон страницы** – Возможность подключить для страницы категории шаблон сайта отличный от основного (*themes/site.php*). Все шаблоны для сайта должны храниться в папке *themes* с расширением *.php* (например, *themes/dizain_so_slajdom.php*). Подробнее в [разделе «Шаблоны сайта»](#). (веб-мастеру и программисту, не меняйте этот параметр, если не уверены в результате!).
- **Шаблон модуля** – Шаблон вывода контента модуля на странице списка товаров в категории (веб-мастеру и программисту, не меняйте этот параметр, если не уверены в результате!).
- **Шаблон модуля для вложенных товаров** – Шаблон вывода контента модуля на странице отдельного товара, вложенного в текущую категорию (веб-мастеру и программисту, не меняйте этот параметр, если не уверены в результате!).

Характеристики

Характеристики

Товары

Характеристики

Категории

Импорт/экспорт

Категория: Все

 [Добавить характеристику](#)

 ☐ Размах крыльев	число
 ☐ Летный потолок	строка
 ☐ Хищность	галочка

Характеристики – это дополнительные параметры, характеризующие товар. Можно добавлять характеристики общие (для всех товаров в пределах одного раздела сайта) или присваивать характеристику одной или нескольким категориям товаров.

Порядок отображения характеристик у товаров можно менять простым перетаскиванием с помощью мыши, удерживая кнопку-ребрышки в начале строки . Кнопка  служит для удаления характеристики. Чтобы прикрепить несколько характеристик к одной категории или, наоборот, открепить от категории, можно воспользоваться групповой операцией.

Редактировать

Название

Тип число

Единица измерения

Категория

Птицы

Звери

Крупные

Мелкие

Сортировка: установить перед [Размах крыльев](#)

Использовать в форме поиска ☒

Характеристики имеют следующие параметры.

Основные:

- **Название** – Имя дополнительной характеристики товара, будет доступно для заполнения при редактировании товара. Параметр имеет разные значения для разных языковых версий сайта.

- **Тип** – Тип дополнительной характеристики. Чтобы назначить характеристику, от которой зависит цена, выберите тип «Список с выбором нескольких значений». Нажмите на ссылку ниже, чтобы почитать подробнее о типах характеристик
Возможные значения:
 - строка,
 - число,
 - дата,
 - дата и время,
 - текстовое поле,
 - галочка,
 - выпадающий список,
 - список с выбором нескольких значений,
 - поле с визуальным редактором,
 - заголовок группы характеристик,
 - файлы,
 - изображения.
- **Максимальное количество добавляемых файлов** – Количество добавляемых файлов. Если значение равно нулю, то форма добавления файлов не выводится. Параметр выводится, если тип характеристики задан как «файлы».
- **Доступные типы файлов (через запятую)** – Параметр выводится, если тип характеристики задан как «файлы».
- **Распознавать изображения** – Позволяет прикрепленные файлы в формате JPEG, GIF, PNG отображать как изображения. Параметр выводится, если тип характеристики задан как «файлы».
- **Размер для большого изображения** – Размер изображения, отображаемый в пользовательской части сайта при увеличении изображения предпросмотра. Параметр выводится, если тип характеристики задан как «файлы» и отмечена опция «Распознавать изображения».
- **Размер для маленького изображения** – Размер изображения предпросмотра. Параметр выводится, если тип характеристики задан как «файлы» и отмечена опция «Распознавать изображения».
- **Использовать анимацию при увеличении изображений** – Параметр добавляет JavaScript код, позволяющий включить анимацию при увеличении изображений. Параметр выводится, если отмечена опция «Распознавать изображения». Параметр выводится, если тип характеристики задан как «файлы» и отмечена опция «Распознавать изображения».
- **Максимальный размер загружаемых файлов** – Параметр показывает максимально допустимый размер загружаемых файлов, установленный в настройках хостинга. Параметр выводится, если тип характеристики задан как «файлы».
- **Генерировать размеры изображений** – Размеры изображений, заданные в модуле «Изображения». Параметр выводится, если тип характеристики задан как «изображение».
- **Доступен к выбору при заказе** – Если отметить, значения этой характеристики пользователь сможет выбрать при покупке товара, например, цвет или размер одежды. Обязательно отметьте данный параметр, для назначения товару зависимой цены или зависимых изображений.
- **Единица измерения** – Параметр выводится, если тип характеристики задан как «число». Параметр имеет разные значения для разных языковых версий сайта.
- **Значения** – Поле появляется для характеристик с типом «галочка», «выпадающий список» и «список с выбором нескольких значений».
- **Отдельная страница для значений** – Если отмечено, на сайте будут сформированы

отдельные страницы, где будут собраны все товары каждого значения характеристик. Например, для характеристики «Производитель» будут сформированы страницы «Sony», «Samsung» и т.д., на каждой из которых будут собраны соответствующие товары.

- **Категории** – Категории, к которым применяется характеристика. Чтобы выбрать несколько категорий, удерживайте CTRL.
- **Использовать в форме поиска** – Если отмечено, данная характеристика товара будет использована при поиске, выводимом тегом show_search.
- **Показывать в списке** – Если отмечено, данная характеристика будет отображаться в списке товаров.
- **Показывать в блоке товаров** – Если отмечено, данная характеристика будет отображаться в блоках товаров, выводимом тегом show_block.
- **Отображать параметры в блоке для сортировки товаров** – Позволяет выводить характеристику в виде ссылки для сортировки товаров по значению характеристики.
- **Описание характеристики** – Поле используется для комментирования характеристик в списке. В пользовательской части поле не используется. Параметр имеет разные значения для разных языковых версий сайта.
- **Сортировка: установить перед** – Изменить положение текущей характеристики среди других характеристик. Например, в фильтре товаров (Администратору сайта).

Какие бывают типы характеристик

- **Строка** – свободная текстовая характеристика товара. Если она назначена категории, у всех товаров появится обычное строковое поле для ввода информации. Использовать её нужно, когда содержимое характеристики небольшое, и у разных товаров оно разное. Например, символный сертификат соответствия.
- **Число** – свободная числовая характеристика товара. Если она назначена категории, у всех товаров появится обычное текстовое поле для ввода чисел. Не принимаются никакие символы, кроме цифр и точки. Использовать её нужно, когда требуется числовая характеристика, и у разных товаров она разная. Например, номер таможенной декларации.
- **Дата** – временная характеристика товара, в формате дд.мм.гггг. Если эта характеристика назначена категории, у товаров появится текстовое поле с календариком для более удобного ввода даты. Используйте эту характеристику, когда товарам требуется дата, например, дата производства.
- **Дата и время** – временная характеристика товара, в формате дд.мм.гггг чч:мм. Характеристика аналогична «Дате», только расширена до времени.
- **Текстовое поле** – свободная текстовая характеристика товара. По свойствам схожа с характеристикой «Строка», но больше по объему. Если она назначена категории, у всех товаров появится обычное текстовое поле для ввода информации. Использовать её нужно, когда содержимое характеристики среднего объема (абзац-два), и у разных товаров оно разное. Например, особенности товара.
- **Галочка** – характеристика для значений Да/Нет, 1/0 или +/- . У товара с этой характеристикой появится для заполнения 2 поля: «да» и «нет». Они позволяют задать отображение параметра в пользовательской части сайта. Если значения пустые, то при показе товара будет отображаться только название характеристики (если стоит значение «Да») и не отображается ничего (для «Нет»).

Пример:

Есть на складе

Если значения заданы, то отображается название характеристики и одно из значений.

Пример:

На складе: есть

На складе: отсутствует.

- **Выпадающий список** – характеристика схожа с характеристикой «Строка», предназначена для небольших текстовых свойств товара. Однако ее отличие в том, что у товара будет сформирован выпадающий список из предустановленных значений характеристики. Это может быть такая характеристика, как, например, цвет, т.е. такое свойство товаров категории, которое изначально ограничено. Заполняете все возможные значения у характеристики, например, для цвета: «Черный», «Белый», «Синий», добавляя их кнопкой , сохраняете характеристику. Затем при заполнении товара у него будет выводиться выпадающий список, где останется только мышкой выбрать нужный цвет.
- **Выпадающий список со множественными значениями** – одна из важнейших характеристик, влияющая на форму и способ заказа товара. По свойствам схожа с характеристикой «Выпадающий список», т.е. состоит из нескольких небольших строковых переменных, однако есть особенности.
 - **Основная особенность** – у товара может быть сразу несколько значений характеристики, например, когда товар бывает разных цветов.
 - **Вторая особенность** – возможность пользователю на сайте выбирать значение характеристики. Например, если товар бывает разных цветов и пользователь должен при заказе цвет выбрать, необходимо отметить галку «Доступен к выбору при заказе»
 - **Третья особенность** – если от характеристики зависит цена. Например, если товар бывает разных объемов, газированная вода 0.5л, 1л, 1.5л – соответственно, разная цена. Необходимо сначала отметить галку «Доступен к выбору при заказе» у характеристики, и заполнить все возможные значения: 0.5, 1, 1.5, используя кнопку . Затем при заполнении товара возле характеристики отметить галку «Влияет на цену». После чего возможно назначать товару несколько цен, в появившихся окошках.
- **Поле с визуальным редактором** – свободная текстовая характеристика товара. По свойствам схожа с характеристиками «Строка» и «Текстовое поле», но она предназначена для самых больших объемов информации, которые к тому же требуется форматировать. Если назначить характеристику категории, то при редактировании поля будет отображаться визуальный редактор. Используйте характеристику, например, для ввода информации о комплектации товара.
- **Заголовок группы характеристик** – служебная характеристика. К товару не относится. Нужна только для того, чтобы визуально разделить характеристики друг от друга. Используется, когда характеристик у товара очень много и их нужно зрительно сгруппировать, как в административной части, так и в пользовательской.
- **Файлы** – характеристика в виде файла. Если к товару нужно прикреплять какие-то дополнительные файлы, например, инструкции пользователя.
- **Изображения** – характеристика в виде изображения, для решения нестандартных задач по оформлению карточки товара. Обычные изображения товара загружаются без характеристики, этот инструмент есть и так в карточке товара. Данная

характеристика нужна для дополнительных иллюстраций, выводимых в дополнительном месте.

Заказы

Заказы

Заказы Форма оформления заказа Отчеты Отложенные Список ожиданий С

[Проверить новые заказы](#)

Статус:

 [Добавить заказ](#)

	11.04.2013 11:56	Заказ № 4	В обработке 37 700 руб.	Ф
	10.04.2013 11:58	Заказ № 3	Новый 33 000 руб.	В

Отображается таблица со всеми заказами, поступающими из пользовательской части сайта. Таблица содержит:

- **Дата** – дата создания заказа;
- **Номер заказа** (в виде ссылки на подробную информацию);
- **Статус** – статусы заказа редактируются в [отдельном интерфейсе](#);
- **Пользователь**, добавивший заказ (в виде ссылки на подробную информацию о пользователе или надпись «без регистрации», если пользователь предпочел не регистрироваться);
- **Дополнительные поля** – группа полей, определенных в части «Оформление заказа».

Кнопка  служит для удаления заказа.

Если оставить страницу заказов открытой, то при поступлении нового заказа в заголовке окна появится мигающее уведомление.

Редактировать

Заказ № 4 ?

Дата 11.04.2013 11:56 ?

Накладная [Сформировать](#)

Наименование товара	Количество	Обычная цена	Цена	Скидка	Сумма
Птицы					
 Пеликан	1	4 700	4700		4 700
 Гусь	1	1 000	1000		1 000

Заказы имеют следующие параметры.

Основные:

- **Заказ №** – Номер заказа.
- **Дата** – Дата создания заказа. Вводится в формате дд.мм.гггг чч:мм.
- **Язык интерфейса** – Фиксируется язык интерфейса, который был при оформлении заказа.
- **Накладная** – Ссылка на товарную накладную. Шаблон редактируется в файле *modules/cart/payment/non_cash/cart.payment.non_cash.memo_tpl.php*.
- **Товары** – Таблица заказанных товаров, сопутствующих услуг. Доступна для редактирования.
- **Способ оплаты** – Список подключенных методов оплаты.
- **Общая скидка.**
- **Способ доставки** – Список подключенных способов доставки.
- **Пользователь** – Если заказал зарегистрированный пользователь.
- **Дополнительные поля** – Группа полей, определенных в части «Форма оформления заказа».
- **Статус** – список подключенных статусов. При смене статуса, у которого действие определено как «оплата, уменьшение количества на складе», делается запись в историю платежей и количество товара уменьшается.

Корзина – это отдельный модуль в пользовательской части сайта (cart). Он устанавливается вместе с модулем «Магазин» и необходим для просмотра корзины и оформления заказа. Сохранение изменений в корзине осуществляются с помощью технологии Ajax, то есть без перезагрузки всей страницы.

При оформлении заказа администратор получает уведомление о поступившем заказе на электронную почту, пользователь получает уведомление об оформлении заказа на электронную почту, и заказ добавляется в базу данных.

Форма оформления заказа

Форму «Оформление заказа» можно дополнить своими полями с помощью конструктора формы.

Форма оформления заказа

Заказы	Форма оформления заказа	Отчеты	Отложенные	Список ожиданий	С
+ Добавить поле					
☐	ФИО или название компании				ст
☐	E-mail				эл
☐	Контактные телефоны (с кодом города)				ст
☐	Город				ст

Порядок отображения полей можно редактировать с помощью мыши и кнопки . Кнопка служит для удаления полей.

Дополнительные поля имеют следующие параметры.

Основные:

- **Название** – Параметр имеет разные значения для разных языковых версий сайта.
- **Значение** – Смысловая нагрузка поля. Возможные значения:
 - Свободное поле,
 - Адрес,
 - Улица,
 - Номер дома,
 - Корпус,
 - Квартира,
 - Подъезд,
 - Этаж,
 - Домофон,
 - Город,
 - Страна,
 - Индекс,
 - Станция метро,
 - Наличие грузового лифта,
 - ФИО,
 - Имя,
 - Фамилия,
 - Отчество,
 - Телефон,

- Дополнительный телефон,
- Электронный адрес для связи,
- Комментарий к адресу.
- **Тип** – Возможные значения:
 - Строка,
 - Число,
 - Дата,
 - Дата и время,
 - Текстовое поле,
 - Галочка,
 - Выпадающий список,
 - Список с выбором нескольких значений,
 - Электронный ящик,
 - Заголовок группы характеристик,
 - Файл,
 - Изображение.
- **Максимальное количество добавляемых файлов** – Количество добавляемых файлов. Если значение равно нулю, то форма добавления файлов не выводится. Параметр выводится, если тип характеристики задан как «файлы».
- **Доступные типы файлов (через запятую)** – Параметр выводится, если тип характеристики задан как «файлы».
- **Распознавать изображения** – Позволяет прикрепленные файлы в формате JPEG, GIF, PNG отображать как изображения. Параметр выводится, если тип характеристики задан как «файлы».
- **Размер для большого изображения** – Размер изображения, отображаемый в пользовательской части сайта при увеличении изображения предпросмотра. Параметр выводится, если тип характеристики задан как «файлы» и отмечена опция «Распознавать изображения».
- **Размер для маленького изображения** – Размер изображения предпросмотра. Параметр выводится, если тип характеристики задан как «файлы» и отмечена опция «Распознавать изображения».
- **Использовать анимацию при увеличении изображений** – Параметр добавляет JavaScript код, позволяющий включить анимацию при увеличении изображений. Параметр выводится, если отмечена опция «Распознавать изображения». Параметр выводится, если тип характеристики задан как «файлы» и отмечена опция «Распознавать изображения».
- **Максимальный размер загружаемых файлов** – Параметр показывает максимально допустимый размер загружаемых файлов, установленный в настройках хостинга. Параметр выводится, если тип характеристики задан как «файлы».
- **Генерировать размеры изображений** – Размеры изображений, заданные в модуле «Изображения». Параметр выводится, если тип характеристики задан как «изображение».
- **Значения** – Появляется для полей с типом «галочка», «выпадающий список» и «список с выбором нескольких значений».
- **Обязательно для заполнения.**
- **Использовать в стандартной форме оформления заказа.**
- **Использовать в форме быстрого заказа.**
- **Позволять редактировать из личного кабинета** – Пользователь сможет установить значение по умолчанию для данного поля из личного кабинета
- **Сортировка: установить перед** – Редактирование порядка следования поля в

форме.

Если выбран тип «выпадающий список» или «список с выбором нескольких значений», то появятся дополнительные поля со значениями. Добавить значение можно нажав  внизу последнего поля. Кнопка  служит для удаления значений.

Отчеты

Отчет о продажах – таблица с перечнем проданных товаров в хронологическом порядке с выводом общей суммы за период.

Отчеты

Заказы	Форма оформления заказа	Отчеты	Отложенные	Список ожиданий	С
Дата: 01.01.2013 - 31.12.2013 23:59 Поиск					
Сегодня Месяц Год					
Дата	Товар	Количество	Сумма		
08.04.2013 18:45	<u>Черепаша Цвет: черный</u>	1	3 200 р		
08.04.2013 18:45	<u>Коршун</u>	1	3 500 р		
08.04.2013 18:45	<u>Пеликан Размер: маленький</u>	1	4 700 р		

Список пожеланий

Список пожеланий – таблица с перечнем товаров, находящемся в списке пожеланий пользователей сайта в хронологическом порядке с выводом общей суммы за период.

Отложенные

Заказы

Форма оформления заказа

Отчеты

Отложенные

Список ожиданий

С

Дата:

Поиск

Сегодня

Месяц

Год

☐ 11.04.2013 11:56

1

Слон, Размер: маленький

☐ 11.04.2013 11:56

1

Рыбка-Карасик, Размер: маленький

☐ Отметить все

Удалить

Применить

На странице: 30

Изменить

Список ожиданий

Список ожиданий – таблица с перечнем товаров, заказанных пользователями через форму «Сообщить, когда появится товар».

Список ожиданий

Заказы	Форма оформления заказа	Отчеты	Отложенные	Список ожиданий	С
--------	-------------------------	--------	------------	-----------------	---

Дата: -

Сегодня Месяц Год

☐	11.04.2013 11:56	erema@ochen_zhdu_kita.ru	Бобр
--------------------------	------------------	--------------------------	----------------------

☐ Отметить все На странице:

Статусы заказа

Статусы заказа

Заказы	Форма оформления заказа	Отчеты	Отложенные	Список ожиданий	С
--------	-------------------------	--------	------------	-----------------	---

☐	<u>Новый</u>	поступление за
☐	<u>В обработке</u>	оплата, уменьш
☐	<u>Отменен</u>	выполнение
☐	<u>Выполнен</u>	отмена заказа

Пользовательский набор статусов заказа.

Статусы имеют следующие свойства:

Основные:

- **Название** – Параметр имеет разные значения для разных языковых версий сайта.
- **Действие** – Действие, при котором статус устанавливается. Действие определяет цвет статуса в панели администрирования. Возможные значения:
 - нет,

- поступление заказа,
 - оплата, уменьшение количества на складе,
 - отмена заказа,
 - выполнение.
- **Сортировка: установить перед** – Редактирование порядка следования поля в списке.

Скидки

Оплата

Скидки	Валюты	Методы оплаты	Способы доставки	Сопутствующие услуги	Ис...
--------	--------	---------------	------------------	----------------------	-------

Дата: -

 [Добавить скидку](#)

☐ 40 %
☐ 15 %
☐ 5 %

Скидки можно установить на весь магазин, на несколько категорий и на несколько товаров.

Добавить новый

Скидка в процентах (%)

Скидка в виде фиксированной суммы

Показывать на сайте ☒

Скидка действует на товары дороже ?

Скидка действует от общей суммы заказа ?

Скидка действует от общей суммы оплаченных заказов ?

Скидки имеют следующие характеристики:

Размер скидки

- **Скидка в процентах (%)** – Если заполнено это поле, скидка будет считаться в процентах от исходной цены.
- **Скидка в виде фиксированной суммы** – Если заполнено это поле, скидка будет вычитаться от исходной цены в виде фиксированной суммы.

На отдельные категории и товары

- **Скидка действует на товары дороже** – Скидка будет применяться только к тем товарам, которые дороже указанной суммы.
- **Объект** – Товары и категории, на которые распространяется скидка. Если не указаны, то скидка считается общей.

На весь заказ

- **Скидка действует от общей суммы заказа** – Скидка начнет действовать когда пользователь наберет в корзину товаров на указанную сумму. Если заполнено, то скидка применяется только в корзине товаров на общую сумму.
- **Накопительная скидка от суммы ранее оплаченных заказов** – Скидка начнет действовать когда пользователь оплатит товаров на указанную сумму. Если заполнено, то скидка применяется только в корзине товаров на общую сумму.

Купоны

- **Код купона** – Пользователь должен активировать на сайте этот код, чтобы получить скидку.

Для отдельных пользователей

- **Группы покупателей** – Скидка будет применяться ко всей группе пользователей.
- **ID** – Если есть пользователи, использующие скидку, то скидка считается персонализированной и другим пользователям не применяется.

Активировать скидку

- **Период действия скидки** – Если выбрать период действия скидки, она будет применяться только в указанное время.
- **Активировать скидку** – Если отметить, скидка будет опубликована на сайте и примениться ко всем товарам, отвечающим условиям выше.

Количество предлагаемых скидок не ограничено. Из нескольких скидок, применяемых для одного товара, выбирается наибольшая.

Валюты

Позволяет создать неограниченное количество валют. Валюты используются для определения цены товара в валюте, отличной от основной. На сайте цены выводятся в основной валюте, пересчет идет по курсу, заданному в модуле.

Валюты

Скидки	Валюты	Методы оплаты	Способы доставки	Сопутствующие услуги	Ис
--------	--------	---------------	------------------	----------------------	----

 [Добавить валюту](#)

☐ [рубль](#)

☐ [евро](#)

☐ [Отметить все](#) [Удалить](#) [Применить](#) На странице: [Изменить](#)

Валюты имеют следующие свойства:

Основные:

- **Название.**
- **Курс к основной валюте** – Все товары на сайте показываются только в основной валюте! Сохраняя в дальнейшем товар в данной валюте, его стоимость будет пересчитываться на сайте по указанному курсу.

Методы оплаты

Методы оплаты

Скидки	Валюты	Методы оплаты	Способы доставки	Сопутствующие услуги	Ис
--------	--------	---------------	------------------	----------------------	----

 [Добавить](#)

 [Наличными курьеру](#)

 [WebMoney](#)

 [Robokassa](#)

 [Банковские платежи](#)

Добавление и редактирование методов оплаты для оформления заказа.

Методы оплаты имеют следующие свойства:

Основные:

- **Название** – Название метода оплаты, выводится на сайте. Параметр имеет разные значения для разных языковых версий сайта.
- **Опубликовать на сайте** – Параметр имеет разные значения для разных языковых версий сайта.
- **Платежная система** – Система безналичной оплаты заказа. Если платежная система не задана, при оформлении заказа сразу перекидывает на страницу завершения заказа. Параметры подключения выдаются платежными системами при одобрении Вашего магазина.
- **Сортировка: установить перед** – Изменить положение текущего метода оплаты среди других методов. В списке методов можно сортировать методы простым перетаскиванием мыши.
- **Описание** – Описание метода оплаты, выводится на сайте в форме заказа. Если отметить «Применить типограф», контент будет отформатирован согласно правилам экранной типографики с помощью [веб-сервиса «Типограф»](#). Параметр имеет разные значения для разных языковых версий сайта.

Для выбора доступны следующие платежные системы:

Webmoney

Дополнительные настройки:

- Webmoney: секретный ключ – ключ, указанный в настройках системы Webmoney;
- Webmoney: кошелек – кошелек, выданный в системе Webmoney.

Параметры настройки Webmoney:

ResultURL: *http://site.ru/cart/payment/webmoney/result/*

SuccessURL: *http://site.ru/cart/payment/webmoney/success/*

FailURL: *http://site.ru/cart/payment/webmoney/fail/*

Банковские платежи

Дополнительные настройки:

- Наименование организации
- ОГРН
- ИНН
- КПП
- Расч. счет
- Банк

- БИК
- Кор. счет
- Адрес
- Руководитель предприятия
- Главный бухгалтер
- КБК
- Сокр. наим. налогового органа
- Код ОКАТО
- НДС
- Печать – поле для загрузки изображения печати.

Robokassa

Дополнительные настройки:

- Robokassa: логин
- Robokassa: пароль 1
- Robokassa: пароль 2

Параметры настройки Robokassa:

ResultURL: *http://site.ru/cart/payment/robokassa/result/*

SuccessURL: *http://site.ru/cart/payment/robokassa/success/*

FailURL: *http://site.ru/cart/payment/robokassa/fail/*

Liqpay

Дополнительные настройки:

- merchant_id
- signature
- result_url
- server_url
- method
- currency

Параметры настройки Liqpay:

ResultURL: *http://site.ru/cart/payment/liqpay/*

ChronoPay

Дополнительные настройки:

- SharedSec
- Product_ID

Параметры настройки ChronoPay:

SuccessURL: *http://site.ru/cart/payment/chronopay/success/*

FailURL: *http://site.ru/cart/payment/chronopay/fail/*

QIWI

Дополнительные настройки:

- Номер терминала
- Пароль

Параметры настройки QIWI:

ResultURL: *http://site.ru/cart/payment/qiwi/result/*

SuccessURL: *http://site.ru/cart/payment/qiwi/success/*

Яндекс.Деньги

Дополнительные настройки:

- scId
- shopId
- shopPassword
- Тестовый режим

Параметры настройки Яндекс.Деньги:

ResultURL: *http://site.ru/cart/payment/yandexmoney/*

Способы доставки

Неограниченное добавление способов доставки.

Способы доставки

Скидки	Валюты	Методы оплаты	Способы доставки	Сопутствующие услуги	Ис...
--------	--------	---------------	------------------	----------------------	-------

 [Добавить](#)

	☐ Курьер
	☐ Самовывоз

☐ [Отметить все](#) [Сделать активным](#)  [Применить](#) На странице: [Изменить](#)

Характеристики способов доставки:

Редактировать

Название

Стоимость от суммы
 от суммы



Показывать на сайте ☒

Сортировка: установить перед [Курьер](#)

Товар доставляется курьером до двери

Описание

Основные:

- **Название** – Название способа доставки, выводится на сайте в форме заказа. Параметр имеет разные значения для разных языковых версий сайта.
- **Стоимость** – Можно указать несколько порогов стоимости доставки в зависимости от суммы заказа. Например, «300 от суммы 0», «150 от суммы 2000», и «0 от суммы 5000».
- **Опубликовать на сайте** – Параметр имеет разные значения для разных языковых версий сайта.
- **Сортировка: установить перед** – Изменить положение текущего способа доставки среди других способов. В списке способов можно сортировать методы простым перетаскиванием мыши.
- **Описание** – Описание способа доставки, выводится в форме заказа. Параметр имеет разные значения для разных языковых версий сайта.

Сопутствующие услуги

Сопутствующие услуги можно выбрать при оформлении заказа.

Сопутствующие услуги

Скидки Валюты Методы оплаты Способы доставки Сопутствующие услуги Иск

+ [Добавить](#)

☐ [Упаковка](#)

☐ [Сертификат](#)

☐ [Ремнабор](#)

1 - 3 Всего 3

Характеристики сопутствующих услуг:

Редактировать

Наименование услуг

Показывать на сайте ☒

Фиксированная стоимость

Процент от суммы заказа

Бесплатно от суммы заказа

Всегда включено в стоимость заказа ☐

Сортировка: установить перед [Упаковка](#)

Основные:

- **Наименование услуг** – Добавляются в корзину при оформлении заказа. Например, «Праздничная упаковка товара» или «Расширенная гарантия на товары». Параметр имеет разные значения для разных языковых версий сайта.
- **Опубликовать на сайте** – Параметр имеет разные значения для разных языковых версий сайта.
- **Фиксированная стоимость.**
- **Процент от суммы заказа** – Стоимость услуги составляет процент от суммы заказа, при этом фиксированная стоимость не учитывается.
- **Бесплатно от суммы заказа** – Стоимость товаров в корзине, при которой данная

услуга осуществляется бесплатно.

- **Всегда включено в стоимость заказа.**
- **Сортировка: установить перед** – Редактирование порядка следования в списке. Поле доступно для редактирования только для услуг, отображаемых на сайте.
- **Описание** – Параметр имеет разные значения для разных языковых версий сайта.

История платежей

Просмотр истории платежей за заданный период времени.

История платежей

Скидки	Валюты	Методы оплаты	Способы доставки	Сопутствующие услуги	Ис
--------	--------	---------------	------------------	----------------------	----

Дата: 12.04.2013 - 12.04.2013 13:39

Дата	Платежная система	Статус	Сумма	Заказ
------	-------------------	--------	-------	-------

Импорт/экспорт (+Я.Маркет)

Импорт/экспорт

Товары	Характеристики	Категории	Импорт/экспорт
--------	----------------	-----------	----------------

 [Добавить описание файла импорта/экспорта](#)

☐ [Импорт категорий](#)

☐ [Импорт товаров](#)

☐ [Отметить все](#) На странице:

Импорт и экспорт товаров использует форма CSV.

Прежде чем начать импорт или экспорт, нужно описать содержание файлов.

Во-первых, нужно добавить новый файл, нажав на ссылку «Добавить описание файла импорта/экспорта».

The screenshot shows a web form titled "Добавить новый" (Add new). It contains several input fields and dropdown menus for configuring file import/export. The fields are: "Название" (Name) with a text input; "Тип" (Type) with a dropdown menu showing "Товары" (Goods); "Удалять неописанные в файле импорта записи" (Delete records not described in the import file) with a checkbox; "Раздел сайта" (Website section) with a dropdown menu showing "Каталог товаров" (Goods catalog); "Категория товаров" (Goods category) with a dropdown menu showing "Все" (All); "Количество строк, выгружаемых за один проход скрипта" (Number of rows exported per script pass) with a text input showing "200"; and "Разделитель данных в строке" (Field separator in row) with a text input showing ":".

Файл имеет следующие характеристики:

Основные:

- **Название** – Краткое описание файла импорта (например, «Импорт товаров», «Импорт цен» и т. д.).
- **Тип** – Возможные значения:
 - Товары,
 - Категории.
- **Удалять неописанные в файле импорта записи** – Если список товаров или категорий в файле является исчерпывающим, то следует отметить эту опцию. Тогда при импорте удалятся ранее занесенные товары или категории, кроме тех, что будут обновлены (определяется по идентификатору).
- **Раздел сайта** – Страница сайта с прикрепленным модулем «Магазин», для которой будет производится импорт.
- **Категория товаров** – Возможность ограничить импорт/экспорт одной категорией магазина.
- **Количество строк, выгружаемых за один проход скрипта** – Время работы скрипта ограничено, из-за чего скрипт может не успеть обработать весь файл за одну итерацию. Поэтому файл обрабатывается частями, величину итерации можно задать этим параметром.
- **Разделитель данных в строке** – Разделитель полей в строке.
- **Обозначать конец строки символом** – Если в строке содержатся символы перевода строки (например, в описании товара), то конец строки должен быть обозначен отдельным символом. Например, КОНЕЦ_СТРОКИ. Не обязательный параметр.
- **Кодировка** – Кодировка данных в файле CSV. Обычно cp1251 или utf8
- **Разделитель данных внутри поля** – Некоторые поля содержат несколько данных в одной ячейки (например, значение характеристики с типом «список с выбором нескольких значений» или изображения), в этом случае данные разделены этим разделителем.

Во-вторых, нужно описать информацию, содержащуюся в файле. Для этого нажимаем на название файла в списке файлов импорта/экспорта или нажимаем кнопку «Сохранить и выйти» при добавлении и редактировании файла импорта.

Импорт товаров [изменить](#)

Товары | Характеристики | Категории | **Импорт/экспорт**

Категория: **Импорт товаров** ▼

Внимание! Перед импортом создайте резервную копию базы данных на случай неудачно должны быть колонки с данными о товарах/категориях в том порядке, который указан

Тип файла для импорта - CSV

[Экспорт в формат CSV](#) Если у Вас есть на сайте товары, Вы можете скачать это файл ка

Поля файла имеют следующие характеристики:

Редактировать

Название

Тип

Выдавать ошибку, если значение не задано ☐

Сортировка: установить перед [Артикул товара](#)

Категория **Импорт товаров** ▼

Основные:

- **Название** – Название поля для импорта, необходимо только для наглядности в списке полей.
- **Тип** – Значение или свойство товара, куда будет импортироваться данное поле. Возможные значения:
 - Идентификатор (уникальный код),
 - Название – поле «Название»,
 - Артикул – используется у товаров,
 - Анонс – поле «Анонс»,
 - Текст – поле «Описание»,

- Ключевые слова, тег Keywords,
- Описание, тег Description,
- Заголовок окна в браузере, тег Title,
- Цена – поле «Цена», используется у товаров,
- Количество – поле «Количество», используется у товаров,
- Категория – идентификатор категории из файла импорта категорий, используется у товаров. Чтобы категория для товара определилась правильно, нужно сначала импортировать категории,
- Пропуск – неиспользуемая информация,
- Родитель – идентификатор родителя (должен соответствовать данным из поля первого типа), используется у категорий,
- Дополнительная характеристика – характеристика товара из конструктора характеристик, используется у товаров,
- Имена изображений – имена изображений через «Разделитель данных внутри поля»,
- Идентификаторы связанных товаров – идентификаторы через «Разделитель данных внутри поля», только для товаров,
- Товар временно отсутствует – значения 1|0|true|false, только для товаров,
- Опубликовать на сайте – значения 1|0|true|false,
- Псевдоссылка – ЧПУ товара/категории,
- Отображать в меню – значения 1|0|true|false,
- Поле «Хит» – значения 1|0|true|false, только для товаров,
- Поле «Новинка» – значения 1|0|true|false, только для товаров,
- Поле «Акция» – значения 1|0|true|false, только для товаров,
- Товар является файлом – значения 1|0|true|false, только для товаров,
- Выгружать в Яндекс Маркет – значения 1|0|true|false,
- Значения полей для Яндекс Маркета – только для товаров,
- Доступ – если доступ ограничен, то идентификаторы типов пользователей, которым дан доступ, через «Разделитель данных внутри поля»,
- Не показывать элемент на карте сайта – значения 1|0|true|false,
- Номер для сортировки – товары и категории сортируются по возрастанию,
- Редактор – id пользователя на сайте,
- Шаблон сайта – файл из папки themes,
- Шаблон модуля (*modules/shop/views/shop.view.шаблон.php*),
- Дата и время начала показа – в формате дд.мм.гггг чч:мм,
- Дата и время окончания показа – в формате дд.мм.гггг чч:мм.
- **Выдавать ошибку, если значение не задано** – При импорте файла выйдет ошибка, если значение поля будет не задано.
- **Использовать в качестве идентификаторов** – Поле выводится только для типов «Идентификатор», «Категория», «Родитель» и «Идентификатор связанных товаров».
 - собственное значение – при первом импорте все товары/категории добавляются в базу, идентификатор запишется в поле `import_id`. При последующем импорте товары/категории будут обновляться по идентификатору `import_id`;
 - идентификатор на сайте – использовать стандартный идентификатор `id`;
 - артикул – только для товаров, только для типов «Идентификатор» и «Идентификатор связанных товаров»;
 - название – только для категорий, только для типов «Категория» и «Родитель».
- **Диапазон значений** – Для полей с типами «Дата и время начала показа» и «Дата и время окончания показа». Помогает исключить ошибки в файле импорта.
- **Дополнительная характеристика** – Список характеристик для поля с типом

«Дополнительная характеристика».

- **Значения списка** – Для дополнительных характеристик с типами «список с выбором нескольких значений» и «выпадающий список». Возможные значения:
 - номер – номер значения списка из таблицы {shop_param_select};
 - название – значение списка, которое видит пользователь.
- **Папка с изображениями для загрузки (относительно корня сайта)** – Только для типа «Имена изображений».
- **Разделитель параметров, влияющих на цену, количества и валюты в пределах одного значения цены/количества** – Только для типов «Цена» и «Количество».
- **Значения параметров, влияющих на цену** – Только для типов «Цена» и «Количество». Возможные значения:
 - номер – номер значения списка из таблицы {shop_param_select};
 - название – значение списка, которое видит пользователь.
- **Указывать количество** – Значение следует сразу за ценой через «Разделитель параметров, влияющих на цену, количества и валюты в пределах одного значения цены/количества», только для типа «Цена».
- **Указывать валюту** – Значение следует сразу за количеством или ценой (если не отмечена опция «Указывать количество») через «Разделитель параметров, влияющих на цену, количества и валюты в пределах одного значения цены/количества», только для типа «Цена».
- **Значение валюты** – Только для типа «Цена». Возможные значения:
 - номер – номер валюты из таблицы {shop_currency};
 - название – название валюты.

Пример:

Свитер имеет 3 цены:

- синий большой - 1200 руб, на складе 3 шт,
- белый большой 100\$, на складе 2 шт
- белый маленький 1000 руб, на складе отсутствует.

Добавляем поле:

- Тип: цена
- Разделитель параметров, влияющих на цену, количества и валюты в пределах одного значения цены/количества: &
- Значения параметров, влияющих на цену: значение
- Указывать количество: да
- Указывать валюту: да

В файле импорта значение поля должно быть следующее:

1200&3&&цвет=синий&размер=большой|100&2&\$&цвет=белый&размер=большой|1000&0&&цвет=белый&размер=маленький.

- **Категория** – Файл импорта.
- **Сортировка: установить после** – Изменить положение текущего поля среди других полей. В списке можно сортировать поля простым перетаскиванием мыши.

Дополнительные:

Когда поля файла описаны, появится ссылка «Экспорт» для скачивания файла экспорта и форма для загрузки файла импорта. При импорте все данные проверяются на валидность, и в случае неправильного формата или некорректных значений выводится лог ошибок.

Примеры импорта товаров и категорий

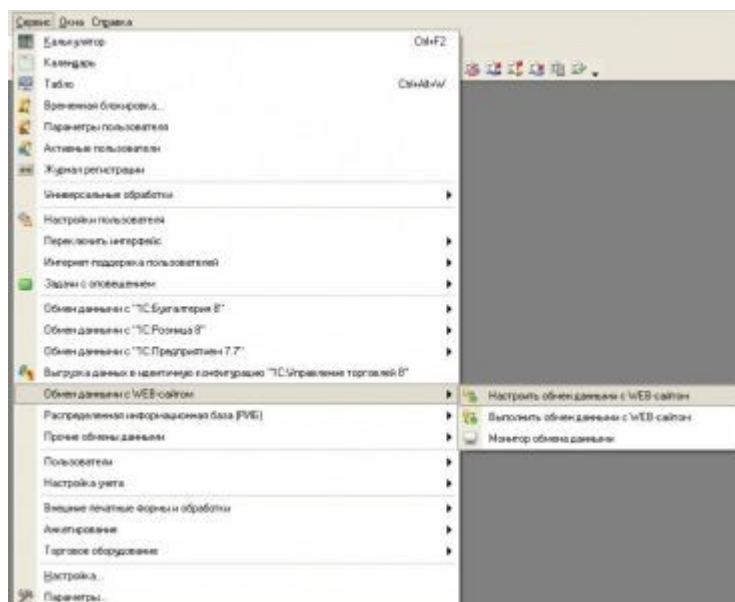
Экспорт товаров в формате YML для Яндекс.Маркет

Выгрузка товаров в формате YML всегда (при подключенной галке «Подключить Яндекс Маркет» в настройках магазина) доступна на сайте по адресу <http://site.ru/modules/shop/shop.yandex.php>

Интеграция с системой 1С:Предприятие

Для синхронизации модуля «Магазин» diafan.CMS и системы 1С:Предприятие 8 в системе 1С выберите пункт меню *Сервис – Обмен данными с web-сайтом – Настроить обмен данными с web-сайтом*.

В открывшемся окне введите: Адрес сайта: http://site.ru/shop/1c/1c_exchange.php
 Пользователь: логин пользователя, у которого есть доступ к редактированию модуля «Магазин»
 Пароль: пароль пользователя, у которого есть доступ к редактированию модуля «Магазин»



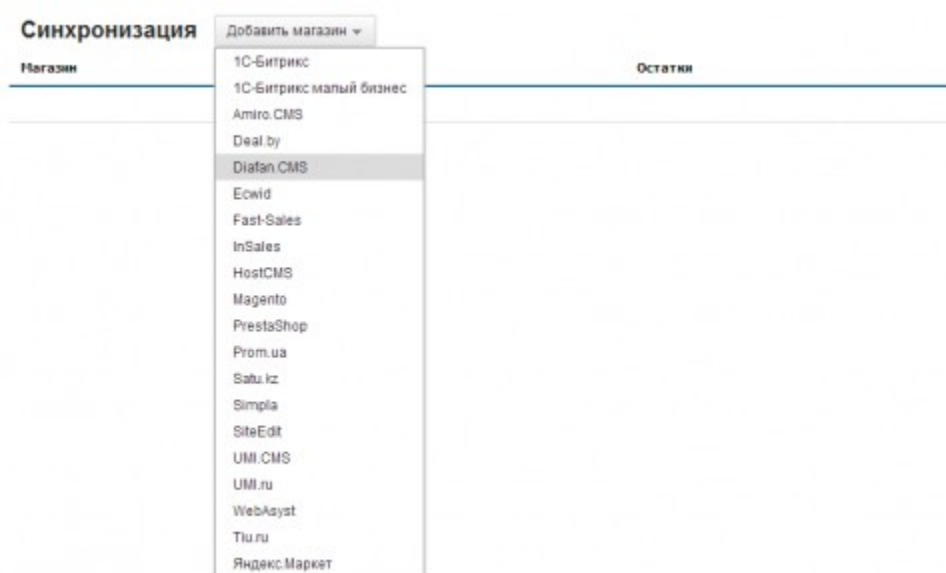
Интеграция разрабатывалась в соответствии с [протоколом обмена между системой «1С:Предприятие» и сайтом](#). Обмен происходит по следующим направлениям:

- выгрузка на сайт торговых предложений – каталогов продукции;
- двусторонний обмен информацией о заказах.

Интеграция с системой МойСклад

Настройка синхронизации diafan.CMS и МойСклад

В сервисе [МойСклад](#) создайте коннектор для обмена (в разделе **Обмен данными | Синхронизация**).



Настройте параметры подключения:

— укажите **Адрес магазина** в настройках синхронизации в МоемСкладе в следующем

формате *http://site.ru/*, заполните поля **Логин** и **Пароль** администратора сайта с правами на сохранение для модуля «Каталог»;

— для проверки правильности введенных данных воспользуйтесь кнопкой **Проверить соединение**;

Настройка Diafan.CMS: diafan-site.ru

Настройка синхронизации Отчет

Сохранить Удалить синхронизацию

* Адрес магазина

* Логин

* Пароль

Проверить соединение

Внимание! Убедитесь, что введенные коды совпадают, иначе синхронизация может привести к потере данных.
[Подробнее...](#)

Заказы

☒ Загружать каждые минут

☐ Копировать номера заказов из магазина

☐ Добавлять к номеру префикс

☐ Резервировать товары

☐ Склад

☐ Организация к

☐ Отправлять статусы заказов в ИМ

Остатки

☒ Выгружать каждые минут

☐ Со склада

☐ Из группы товаров

Цены

Выгружать остатки с учетом характеристик ☐

— запустите коннектор в настройках синхронизации в МоемСкладе (установите флажок **Загружать/Выгружать каждые [XX] минут**).

Параметры коннектора diafan.CMS и МойСклад

Заказы

Загрузить сейчас — запустить загрузку заказов немедленно.

Загружать каждые [XX] минут — если флажок установлен, то МойСклад будет автоматически забирать заказы из интернет-магазина. Период, как часто это будет происходить, указывается в поле (минимальное значение 5 минут).

Копировать номера заказов из магазина — если флажок установлен, то заказ будет создан с номером, который пришел из интернет-магазина. Если флажок снят, то номер будет назначен МоимСкладом (номер последнего созданного заказа + 1).

Добавлять к номеру префикс — символы, которые вы укажете в этом поле, будут добавлены перед номером заказа (Пример: из интернет-магазина пришел заказ с номером 100, мы указали префикс «ИнМаг-», значит будет создан заказ с номером «ИнМаг-100»).

Резервировать товары — если флажок установлен, то товары из приходящих заказов будут автоматически ставиться в резерв.

Склад — склад, который будет подставлен в заказ.

Организация — ваша организация, которая будет подставлена в заказ.

Отправлять статусы заказов в ИМ — если флажок установлен, состояние заказа в МоемСкладе будет передаваться в интернет-магазин при синхронизации.

Остатки

Выгрузить сейчас — запустить выгрузку остатков товаров немедленно.

Выгружать каждые [XX] минут — если флажок установлен, то МойСклад будет автоматически выгружать информацию об остатках товаров в интернет-магазин. Период, как часто будет происходить выгрузка, указывается в поле (минимальное значение 30 минут).

Со склада — склад, с которого будут выгружаться остатки товаров. Если склад не указан, то будут выгружены остатки по всем складам.

Из группы товаров — группа товаров, для товаров из которой будут выгружаться остатки. Если группа не указана, то будут выгружены остатки для всех товаров.

Выгружать остатки с учетом характеристик — все остатки по имеющейся номенклатуре товаров будут выгружены с учетом характеристик.

Внимание! После изменения настроек не забудьте их сохранить.

Настройки модуля

Можно сохранять разные настройки модуля для разных страниц, к которым прикреплен модуль.

Настройки

Основные настройки	Изображения	Яндекс Маркет	Сообщения и уведомления
Запретить покупать товары	☐	?	
Покупать могут только зарегистрированные	☐	?	
Учитывать остатки товаров на складе	☐	?	
Страница окончания оформления заказа	<u>Заказ оформлен</u>  ?		
Количество товаров на странице	10		

Основные настройки

- **Запретить покупать товары** — Чтобы запретить пользователям покупать товары и использовать сайт как каталог товаров без кнопки «Купить» и без «Корзины», удалите или деактивируйте страницу сайта с прикрепленным модулем «Корзина». Чтобы

вернуть возможность продажи товаров, в системе (в модуле «Страницы сайта») должна существовать страница с подключенным модулем «Корзина».

- **Покупать могут только зарегистрированные** – Если отмечена, кнопка купить будет появляться только для авторизованных пользователей.
- **Учитывать остатки товаров на складе** – Если отмечена, у каждого товара появится поле с количеством товара и необходимо будет указывать количество товара на складе. Товар с остатком 0 нельзя купить, кнопка «Купить» будет заменена на сообщение «Товар отсутствует». При покупке товара и в выполнении заказа (статус заказа с товаром «Выполнен») количество будет минусоваться.
- **Отключить сравнение товаров** – Если отмечена, у товаров не будет кнопки «Сравнить».
- **Разрешать покупать товары без цены** – Если отмечена, товар с нулевой ценой можно положить в корзину.
- **Разрешить «Заказать в один клик»** – На странице товара появится форма быстрой покупки товара. Поля для формы выводятся опцией «Использовать в форме быстрого заказа».
- **Количество товаров на странице** – Количество одновременно выводимых товаров в списке.
- **Страница окончания оформления заказа** – Страница сайта, на которую попадает покупатель после успешного оформления заказа.
- **Включить возможность продажи файлов** – Если отмечена, возможно продавать файлы. Тогда товаром можно сделать загружаемый файл, который будет храниться в защищенном месте и после оплаты пользователем доступен ему по уникальной ссылке, «живущей» один час.
- **Доступные типы файлов (через запятую)** – Параметр появляется, если отмечена опция «Включить возможность продажи файлов».

Пример:

doc, gif, jpg, mpg, pdf, png, txt, zip

- **Использовать категории** – Позволяет включить/отключить категории товаров.
- **Количество товаров в списке категорий** – Для первой страницы магазина, где выходят по несколько товаров из всех категорий. Параметр появляется, если отмечена опция «Использовать категории».
- **Количество товаров в списке вложенной категории** – Для первой страницы модуля и для страницы категории. Параметр появляется, если отмечена опция «Использовать категории».
- **Показывать товары подкатегорий** – Если отмечена, в списке товаров категории будут отображаться товары из всех вложенных категорий. Параметр появляется, если отмечена опция «Использовать категории».
- **Подключить счетчик просмотров** – Позволяет считать количество просмотров отдельного товара.
- **Выводить счетчик на сайте** – Позволяет вывести на сайте количество просмотров отдельного товара. Параметр выводится, если отмечена опция «Счетчик просмотров». Параметр появляется, если отмечена опция «Подключить счетчик просмотров».
- **Название основной валюты** – Название основной валюты сайта. Для добавления дополнительных валют, воспользуйтесь модулем "Оплата" Параметр имеет разные значения для разных языковых версий сайта.
- **Формат цены** – Возможность указать формат отображения цены (количество знаков после запятой, разделитель целых, разделитель десятков).

- **Искать по цене** – Параметр позволяет выводить в блоке поиска по товарам поиск по цене.
- **Искать по описанию** – Параметр позволяет выводить в блоке поиска по товарам поиск по описанию.
- **Искать по названию** – Параметр позволяет выводить в блоке поиска по товарам поиск по названию.
- **Искать по артикулу** – Параметр позволяет выводить в блоке поиска по товарам поиск по артикулу.
- **Искать товары по акции** – Параметр позволяет выводить в блоке поиска по товарам поиск товаров, участвующих в акциях (у товара отмечена опция «Акция»).
- **Искать по новинкам** – Параметр позволяет выводить в блоке поиска по товарам поиск новинок (у товара отмечена опция «Новинок»).
- **Искать по хитам** – Параметр позволяет выводить в блоке поиска по товарам поиск хитов (у товара отмечена опция «Хит»).
- **Подключить комментарии к товарам** – Подключение модуля «Комментарии». Параметр не будет включен, если модуль «Комментарии» не установлен. Подробности см. в разделе [модуль Комментарии](#).
- **Показывать комментарии к категориям** – Подключение модуля «Комментарии» к категориям товаров. Параметр не будет включен, если модуль «Комментарии» не установлен. Подробности см. в разделе [модуль Комментарии](#).
- **Подключить теги к товарам** – Подключение модуля «Теги». Параметр не будет включен, если модуль «Теги» не установлен. Подробности см. в разделе [модуль «Теги»](#).
- **Подключить рейтинг товаров** – Подключение модуля «Рейтинг». Параметр не будет включен, если модуль «Рейтинг» не установлен. Подробности см. в разделе [модуль «Рейтинг»](#).
- **Подключить рейтинг к категориям** – Подключение модуля «Рейтинг» к категориям. Параметр не будет включен, если модуль «Рейтинг» не установлен. Подробности см. в разделе [модуль «Рейтинг»](#).
- **Подключить перелинковку** – Отображение перелинковки в модуле. Подробности см. в разделе [модуль «Перелинковка»](#).
- **В блоке похожих товаров связь двусторонняя** – Если отметить, то при назначении товару А похожего товара Б, у товара Б автоматически станет похожим товар А.
- **Шаблон для автоматического генерирования Title** – Если шаблон задан, то заголовок автоматически генерируется по шаблону. В шаблон можно добавить:
 - %title – заданный заголовок,
 - %name – название,
 - %category – название категории,
 - %parent_category – название категории верхнего уровня (SEO-специалисту).
 Параметр имеет разные значения для разных языковых версий сайта.
- **Шаблон для автоматического генерирования Keywords** – Если шаблон задан, то поле *Keywords* автоматически генерируется по шаблону. В шаблон можно добавить:
 - %keywords – заданные ключевые слова,
 - %name – название,
 - %category – название категории,
 - %parent_category – название категории верхнего уровня (SEO-специалисту).
 Параметр имеет разные значения для разных языковых версий сайта.
- **Шаблон для автоматического генерирования Description** – Если шаблон задан, то поле *Description* автоматически генерируется по шаблону. В шаблон можно добавить:
 - %descr – заданное описание,
 - %name – название,

- %category – название категории,
- %parent_category – название категории верхнего уровня (SEO-специалисту).

Параметр имеет разные значения для разных языковых версий сайта.

- **Шаблон для автоматического генерирования Title для категории** – Если шаблон задан, то заголовок автоматически генерируется по шаблону. В шаблон можно добавить:

- %title – заданный заголовок,
- %name – название категории,
- %parent – название категории верхнего уровня,

- %page – страница (текст можно поменять в интерфейсе «Языки сайта» – «Перевод интерфейса») (SEO-специалисту).

Параметр имеет разные значения для разных языковых версий сайта. Параметр появляется, если отмечена опция «Использовать категории».

- **Шаблон для автоматического генерирования Keywords для категории** – Если шаблон задан, то поле *Keywords* автоматически генерируется по шаблону. В шаблон можно добавить:

- %keywords – заданные ключевые слова,
- %name – название категории,
- %parent – название категории верхнего уровня (SEO-специалисту).

Параметр имеет разные значения для разных языковых версий сайта. Параметр появляется, если отмечена опция «Использовать категории».

- **Шаблон для автоматического генерирования Description для категории** – Если шаблон задан, то поле *Description* автоматически генерируется по шаблону. В шаблон можно добавить %descr - заданное описание, %name - название категории, %parent - название категории верхнего уровня. Параметр имеет разные значения для разных языковых версий сайта. Параметр появляется, если отмечена опция «Использовать категории».
- **Шаблон для списка элементов** – По умолчанию *modules/shop/views/shop.view.list.php*. Параметр для разработчиков! Не устанавливайте, если не уверены в результате.
- **Шаблон для первой страницы модуля (если подключены категории)** – По умолчанию *modules/shop/views/shop.view.first_page.php*. Параметр для разработчиков! Не устанавливайте, если не уверены в результате.
- **Шаблон для страницы элемента** – По умолчанию, *modules/shop/views/shop.view.id.php*. Параметр для разработчиков! Не устанавливайте, если не уверены в результате.
- **Шаблон для списка элементов с одинаковой характеристикой** – По умолчанию, *modules/shop/views/shop.view.list.php*. Параметр для разработчиков! Не устанавливайте, если не уверены в результате.
- **Шаблон для сравнения товаров** – По умолчанию, *modules/shop/views/shop.view.compare.php*. Параметр для разработчиков! Не устанавливайте, если не уверены в результате.
- **Шаблон для поиска элементов** – По умолчанию, *modules/shop/views/shop.view.list.php*. Параметр для разработчиков! Не устанавливайте, если не уверены в результате.
- **Индексирование для карты сайта** – При изменении настроек, влияющих на отображение страницы, модуль автоматически переиндексируется для карты сайта *sitemap.xml*.

Изображения

- **Использовать изображения** – Позволяет включить/отключить загрузку изображений к товарам.
- **Генерировать размеры изображений** – Размеры изображений, заданные в модуле «Изображения» и тег латинскими буквами для подключения изображения на сайте. Обязательно должны быть заданы два размера: превью изображения в списке товаров (тег medium) и полное изображение (тег large). Если задан дополнительный вариант, помеченный тегом preview, то на странице товара будет выводиться уменьшенные изображения (preview), при нажатии на которые обновлится основное изображение товара (medium).
- **Отображение изображений в списке** – Параметр принимает значения:
 - нет (отключает отображение изображений в списке);
 - показывать одно изображение;
 - показывать все изображения. Параметр выводится, если отмечена опция «Использовать изображения».
- **Использовать изображения для категорий** – Позволяет включить/отключить загрузку изображений к категориям.
- **Генерировать размеры изображений для категорий** – Размеры изображений, заданные в модуле «Изображения» и тег латинскими буквами для подключения изображения на сайте. Обязательно должны быть заданы два размера: превью изображения в списке категорий (тег medium) и полное изображение (тег large). Параметр выводится, если отмечена опция «Использовать изображения для категорий».
- **Отображение изображений в списке категорий** – Параметр принимает значения:
 - нет (отключает отображение изображений в списке);
 - показывать одно изображение;
 - показывать все изображения. Параметр выводится, если отмечена опция «Использовать изображения для категорий».
- **Использовать анимацию при увеличении изображений** – Параметр добавляет JavaScript код, позволяющий включить анимацию при увеличении изображений. Параметр выводится, если отмечена опция «Использовать изображения».
- **Максимальный размер загружаемых файлов** – Параметр показывает максимально допустимый размер загружаемых файлов, установленный в настройках хостинга. Параметр выводится, если отмечена опция «Использовать изображения».
- **Применить настройки ко всем ранее загруженным изображениям** – Позволяет перекодировать размер уже загруженных изображений. Кнопка необходима, если изменены настройки размеров изображений. Параметр выводится, если отмечена опция «Использовать изображения».

Яндекс Маркет

- **Подключить Яндекс Маркет** – Если отметить, по адресу <http://www.site.ru/modules/shop/shop.yandex.php> будет активен файл с импортом товаров для системы «Яндекс.Маркет» в формате YML. Все подробности и требования к магазинам смотрите на [сайте «Яндекс Маркет»](#).
- **Короткое название магазина** – Название магазина для системы «Яндекс Маркет». Не должно содержать более 20 символов. Нельзя использовать слова, не имеющие отношения к наименованию магазина («лучший», «дешевый»), указывать номер телефона и т. п. Название магазина, должно совпадать с фактическим названием

магазина, которое публикуется на сайте). Параметр появляется, если отмечена опция «Подключить Яндекс Маркет».

- **Валюта** – Валюта для системы «Яндекс Маркет». Возможные значения:
 - RUR,
 - USD,
 - EUR,
 - UAH,
 - BYR,
 - KZT.

Параметр появляется, если отмечена опция «Подключить Яндекс Маркет».

- **Выгружать категории в Яндекс.Маркет** – Позволяет выбрать какие категории выгружать в «Яндекс Маркет»: все или только помеченные (появляется галочка при редактировании категории). Возможные значения:
 - все,
 - только помеченные.

Параметр появляется, если отмечена опция «Подключить Яндекс Маркет».

- **Выгружать товары в Яндекс.Маркет** – Позволяет выбрать какие товары выгружать в «Яндекс Маркет»: все или только помеченные (появляется галочка при редактировании товара). Возможные значения:
 - все,
 - только помеченные.

Параметр появляется, если отмечена опция «Подключить Яндекс Маркет».

- **Основная ставка** – Смотрите [инструкцию «Яндекс Маркет»](#). Параметр появляется, если отмечена опция «Подключить Яндекс Маркет».
- **Ставка для карточек** – Смотрите [инструкцию «Яндекс Маркет»](#). Параметр появляется, если отмечена опция «Подключить Яндекс Маркет».
- **Подключить Яндекс Быстрый заказ** – Позволяет заполнять форму оформления заказа данными, предзаполненными в [системе «Яндекс Быстрый заказ»](#).

Сообщения и уведомления

- **E-mail, указываемый в обратном адресе пользователю** – Возможные значения:
 - e-mail, указанный в параметрах сайта;
 - другой (при выборе этого значения появляется дополнительное поле **впишите e-mail**).
- **Тема письма пользователю о поступлении товара** – Можно добавлять:
 - %title – название сайта,
 - %url – адрес сайта (например, site.ru).

Параметр имеет разные значения для разных языковых версий сайта.

Пример:

Вы оформили заказ на сайте %title (%url)

- **Сообщение пользователю о поступлении товара** – Можно добавлять:
 - %title – название сайта,
 - %url – адрес сайта (например, site.ru),
 - %good – название товара,
 - %link – ссылка на товар.

Параметр имеет разные значения для разных языковых версий сайта.

Пример:

Здравствуйте!
Товар %good поступил на склад.

- **Сообщение о совершенном заказе перед оплатой** – Сообщение, получаемое пользователем по окончании оформления заказа. Параметр имеет разные значения для разных языковых версий сайта.

Пример:

Спасибо за Ваш заказ! В ближайшее время мы с Вами свяжемся для подтверждения заказа.

- **Описание платежа** – Используется платежными системами. Параметр имеет разные значения для разных языковых версий сайта.

Пример:

Oplata tovarov.

- **Платеж успешно принят** – Сообщение, которое увидит пользователь, если платеж успешно принят платежной системой.

Пример:

Спасибо, платеж успешно принят. В ближайшее время мы с Вами свяжемся для уточнения деталей заказа.

- **Платеж не принят** – Сообщение, которое увидит пользователь, если платеж не принят платежной системой.

Пример:

Извините, платеж не прошел.

- **Тема письма пользователю о новом заказе** – Можно добавлять:

- %title – название сайта,
- %url – адрес сайта (например, site.ru).

Параметр имеет разные значения для разных языковых версий сайта.

Пример:

Вы оформили заказ на сайте %title (%url)

- **Сообщение пользователю о новом заказе** – Можно добавлять:

- %title – название сайта,
- %url – адрес сайта (например, site.ru),
- %order – таблица заказа,
- %payment – способ оплаты,
- %message – поля формы «Оформление заказа»,
- %id – номер заказа.

Параметр имеет разные значения для разных языковых версий сайта.

Пример:

Здравствуйте!
Вы оформили заказ на сайте %title (%url):
Номер заказа: %id
%order
Способ оплаты: %payment
%message
Спасибо за Ваш заказ! В ближайшее время мы с Вами свяжемся для подтверждения заказа.

- **Тема письма пользователю об изменении статуса заказа** – Можно добавлять:
 - %title – название сайта,
 - %url – адрес сайта (например, site.ru).

Параметр имеет разные значения для разных языковых версий сайта.

Пример:

Статус заказа изменен

- **Сообщение пользователю об изменении статуса заказа** – Можно добавлять:
 - %title – название сайта,
 - %url – адрес сайта (например, site.ru),
 - %order – номер заказа,
 - %status – новый статус.

Параметр имеет разные значения для разных языковых версий сайта.

Пример:

Здравствуйте!
Статус заказ №%order изменен на «%status».

- **Тема письма администратору о новом заказе** – Можно добавлять:
 - %title – название сайта,
 - %url – адрес сайта (например, site.ru).

Пример:

%title (%url). Новый заказ

- **Текст письма администратору о новом заказе** – Можно добавлять:
 - %title – название сайта,
 - %url – адрес сайта (например, site.ru),
 - %order – таблица заказа,
 - %payment – способ оплаты,
 - %message – поля формы «Оформление заказа»,
 - %id – номер заказа.

Пример:

Здравствуйте, администратор сайта %title (%url)!
На сайте появился новый заказ номер %id: %order
Способ оплаты: %payment
%message

- **E-mail для уведомлений администратора** – Возможные значения:
 - e-mail, указанный в параметрах сайта;
 - другой (при выборе этого значения появляется дополнительное поле **впишите e-mail**).
- **Уведомлять о поступлении новых заказов по SMS** – Возможность отправлять SMS администратору при поступлении заказа. Параметр можно подключить, если в [Параметрах сайта](#) настроены SMS-уведомления.

- **Номер телефона в федеральном формате** – Номер телефона для SMS-уведомлений администратора о новом заказе. Параметр появляется, если отмечена опция «Уведомлять о поступлении новых заказов по SMS».
- **Сообщение для уведомлений** – Текст сообщения для SMS-уведомлений администратора о новом заказе. Не более 800 символов. Параметр появляется, если отмечена опция «Уведомлять о поступлении новых заказов по SMS».
- **Сообщение пользователю о купленных файлах** – Можно добавлять:
 - %title – название сайта,
 - %url – адрес сайта (например, site.ru),
 - %files – ссылки на скачивание файлов,
 - %id – номер заказа.

Пример:

Здравствуйте!
Вы оформили заказ на сайте %title (%url):

Номер заказа: %id
Файлы можно скачать по ссылкам в течении часа: %files

Спасибо за Ваш заказ!

Шаблонные теги

Для работы с модулем «Магазин» служат следующие [шаблонные теги](#):

show_add_coupon – выводит форму активирования купона на скидку, если неактивированный купон есть в системе, пользователь авторизован и у него не активирован другой купон.

Атрибуты:

— **template** – шаблон тега (файл *modules/shop/views/shop.view.show_add_coupon_template.php*; по умолчанию шаблон *modules/shop/views/shop.view.show_add_coupon.php*).

Пример:

```
<insert name="show_add_coupon" module="shop">
```

выведет форму активирования купона

show_block module="cart" – выводит информацию о заказанных товарах, т. н. корзину.

Атрибуты:

— **template** – шаблон тега (файл *modules/cart/views/cart.view.show_block_template.php*; по умолчанию шаблон *modules/cart/views/cart.view.show_block.php*).

Пример:

```
<insert name="show_block" module="cart">
```

выведет информацию о корзине

show_block – выводит несколько товаров из каталога.

Атрибуты:

- **count** – количество выводимых товаров (по умолчанию 3);
- **site_id** – страницы, к которым прикреплен модуль. Идентификаторы страниц перечисляются через запятую. По умолчанию выбираются все страницы;
- **cat_id** – категории товаров, если в настройках модуля отмечено «Использовать категории». Идентификаторы категорий перечисляются через запятую. По умолчанию категория не учитывается, выводятся все товары;
- **images** – количество изображений, прикрепленных к товару;
- **images_variation** – тег размера изображений, задается в настройках модуля;
- **sort** – сортировка товаров: по умолчанию как на странице модуля, **date** – по дате, **rand** – в случайном порядке, **price** – по цене, **sale** – по количеству продаж;
- **only_shop** – выводить блок только на странице, к которой прикреплен модуль «Магазин»;
- **hits_only** – выводить только товары с пометкой «Хит»;
- **action_only** – выводить только товары с пометкой «Акция»;
- **new_only** – выводить только товары с пометкой «Новинка»;
- **param** – значения дополнительных характеристики;

Пример:

Товары обладают следующими характеристиками:

- цвет – выпадающий список, номер 3;
- высота – число, номер 10;
- наличие аналогов – галочка, номер 16.

Значит значение атрибута *param3=5&3=6&10>12&16=0* расшифровывается как товары красного и синего цвета (5 и 6 номер), высотой более 12, не имеющие аналогов.

— **template** – шаблон тега (файл *modules/shop/views/shop.view.show_block_template.php*; по умолчанию шаблон *modules/shop/views/shop.view.show_block.php*);

— **discount_only** – выводить только товары, на которые действует скидка.

Пример:

```
<insert name="show_block" module="shop">
```

выведет 3 последних товара из магазина

```
<insert name="show_block" module="shop" count="5" sort="rand">
```

выведет 5 случайных товаров из магазина

```
<insert name="show_block" module="shop" sort="price" count="4" cat_id="12">
```

выведет 4 самых дешевых товаров из рубрики №12 магазина

show_block_order_rel – товары, которые обычно покупают с текущим товаром.

Атрибуты:

- **count** – количество выводимых товаров (по умолчанию 3);
- **images** – количество изображений, прикрепленных к товару;
- **images_variation** – тег размера изображений, задается в настройках модуля;
- **template** – шаблон тега (файл *modules/shop/views/shop.view.show_block_order_rel_template.php*; по умолчанию шаблон *modules/shop/views/shop.view.show_block_order_rel.php*).

Пример:

```
<insert name="show_block_order_rel" module="shop">
```

выведет 3 товара, которые обычно покупают с текущим товаром

show_block_rel – на странице товара выводит похожие товары. По умолчанию связи между товарами являются односторонними, это можно изменить, отметив опцию «В блоке похожих товаров связь двусторонняя» в настройках модуля.

Атрибуты:

- **count** – количество выводимых товаров (по умолчанию 3);
- **images** – количество изображений, прикрепленных к товару;
- **images_variation** – тег размера изображений, задается в настройках модуля;
- **template** – шаблон тега (файл *modules/shop/views/shop.view.show_block_rel_template.php*; по умолчанию шаблон *modules/shop/views/shop.view.show_block_rel.php*).

modules/shop/views/shop.view.show_block_rel.php).

Пример:

```
<insert name="show_block_rel" module="shop">
```

выведет 3 товара, прикрепленные к текущему товару

show_search – выводит форму поиска товаров. Если для категорий прикреплены дополнительные характеристики, то поиск по ним производится только на странице категории. Поиск по обязательным полям подключается в настройках модуля (опции «Искать по цене», «Искать по описанию», «Искать по названию», «Искать по артикулу», «Искать товары по акции», «Искать новинки», «Искать хиты»). Если в форму поиска выведены характеристики с типом «выпадающий список» и «список с выбором нескольких значений», то значения характеристик, которые не найдут ни один товар, в форме поиска не выведутся.

Атрибуты:

— **only_shop** – выводить форму поиска только на странице магазина;

— **site_id** – страницы, к которым прикреплен модуль. Идентификаторы страниц перечисляются через запятую. По умолчанию выбираются все страницы. Если выбрано несколько страниц сайта, то в форме поиска появляется выпадающий список по выбранным страницам;

— **cat_id** – категории товаров, если в настройках модуля отмечено «Использовать категории». Идентификаторы категорий перечисляются через запятую. Можно указать значение **current**, тогда поиск будет осуществляться по текущей (открытой) категории магазина или по всем категориям, если ни одна категория не открыта. Если выбрано несколько категорий, то в форме поиска появится выпадающий список категорий магазина, который будет подгружать прикрепленные к категориям характеристики. Можно указать значение **all**, тогда поиск будет осуществляться по всем категориям товаров и в форме будут участвовать только общие характеристики;

— **ajax** – подгружать результаты поиска без перезагрузки страницы. Результаты подгружаются только если открыта страница со списком товаров, иначе поиск работает обычным образом;

— **template** – шаблон тега (файл *modules/shop/views/shop.view.show_search_template.php*; по умолчанию шаблон *modules/shop/views/shop.view.show_search.php*).

Пример:

```
<insert name="show_search" module="shop">
```

выведет форму поиска по каталогу товаров

Файлы

1. *modules/cart/cart.php* – контроллер модуля «Корзина товаров, оформление заказа»;
2. *modules/cart/cart.ajax.php* – обработка Ajax-запросов;
3. *modules/cart/cart.inc.php* – подключение модуля «Корзина товаров, оформление заказа»;
4. *modules/cart/cart.js* – JS-сценарий модуля «Корзина товаров, оформление заказа»;
5. *modules/cart/cart.model.php* – модель модуля «Корзина товаров, оформление заказа»;
6. *modules/cart/cart.payment.php* – работа с платежными системами;
7. *modules/cart/payment/chronopay/cart.payment.chronopay.php* – работа с платежной системой «ChronoPay», callback;
8. *modules/cart/payment/chronopay/cart.payment.chronopay.admin.php* – настройки платежной системы «ChronoPay» для административного интерфейса;
9. *modules/cart/payment/chronopay/cart.payment.chronopay.model.php* – формирует данные для формы платежной системы «ChronoPay»;
10. *modules/cart/payment/chronopay/cart.payment.chronopay.view.php* – шаблон платежа через систему «ChronoPay»;
11. *modules/cart/payment/liqpay/cart.payment.liqpay.php* – работа с платежной системой «Liqpay»;
12. *modules/cart/payment/liqpay/cart.payment.liqpay.admin.php* – настройки платежной системы «Liqpay» для административного интерфейса;
13. *modules/cart/payment/liqpay/cart.payment.liqpay.model.php* – формирует данные для формы платежной системы «Liqpay»;
14. *modules/cart/payment/liqpay/cart.payment.liqpay.view.php* – шаблон платежа через систему «Liqpay»;
15. *modules/cart/payment/non_cash/cart.payment.non_cash.php* – платежная квитанция на оплату;
16. *modules/cart/payment/non_cash/cart.payment.non_cash.admin.php* – настройки платежной системы «Банковские платежи» для административного интерфейса;
17. *modules/cart/payment/non_cash/cart.payment.non_cash.fl_tpl.php* – шаблон квитанции для физического лица;
18. *modules/cart/payment/non_cash/cart.payment.non_cash.memo_tpl.php* – шаблон товарного чека;
19. *modules/cart/payment/non_cash/cart.payment.non_cash.model.php* – формирует данные для страницы оплаты безналичным платежом;
20. *modules/cart/payment/non_cash/cart.payment.non_cash.num2str.php* – перевод чисел в слова;

21. *modules/cart/payment/non_cash/cart.payment.non_cash.ul_tpl.php* – шаблон квитанции для юридического лица;
22. *modules/cart/payment/non_cash/cart.payment.non_cash.view.php* – шаблон безналичного платежа;
23. *modules/cart/payment/qiwi/cart.payment.qiwi.php* – обработка данных, полученных от системы QIWI;
24. *modules/cart/payment/qiwi/cart.payment.qiwi.admin.php* – настройки платежной системы QIWI для административного интерфейса;
25. *modules/cart/payment/qiwi/cart.payment.qiwi.model.php* – формирует данные для формы платежной системы QIWI;
26. *modules/cart/payment/qiwi/cart.payment.qiwi.view.php* – шаблон платежа через систему QIWI;
27. *modules/cart/payment/robokassa/cart.payment.robokassa.php* – обработка данных, полученных от системы Robokassa;
28. *modules/cart/payment/robokassa/cart.payment.robokassa.admin.php* – настройки платежной системы Robokassa для административного интерфейса;
29. *modules/cart/payment/robokassa/cart.payment.robokassa.model.php* – формирует данные для формы платежной системы Robokassa;
30. *modules/cart/payment/webmoney/cart.payment.webmoney.php* – обработка данных, полученных от системы WebMoney;
31. *modules/cart/payment/webmoney/cart.payment.webmoney.admin.php* – настройки платежной системы WebMoney для административного интерфейса;
32. *modules/cart/payment/webmoney/cart.payment.webmoney.model.php* – формирует данные для формы платежной системы WebMoney;
33. *modules/cart/payment/webmoney/cart.payment.webmoney.view.php* – шаблон платежа через систему WebMoney;
34. *modules/cart/payment/yandexmoney/cart.payment.yandexmoney.php* – работа с платежной системой Яндекс.Деньги;
35. *modules/cart/payment/yandexmoney/cart.payment.yandexmoney.admin.php* – настройки платежной системы Яндекс.Деньги для административного интерфейса;
36. *modules/cart/payment/yandexmoney/cart.payment.yandexmoney.model.php* – формирует данные для формы платежной системы Яндекс.Деньги;
37. *modules/cart/payment/yandexmoney/cart.payment.yandexmoney.view.php* – шаблон платежа через систему Яндекс.Деньги;
38. *modules/cart/views/cart.view.form.php* – шаблон формы редактирования корзины товаров, оформления заказа;
39. *modules/cart/views/cart.view.images.php* – шаблон прикрепленных изображений;

- 40. *modules/cart/views/cart.view.info.php* – шаблон информации о товарах в корзине;
- 41. *modules/cart/views/cart.view.one_click.php* – шаблон форма оформления заказа в один клик;
- 42. *modules/cart/views/cart.view.payment.php* – шаблон формы платежной системы;
- 43. *modules/cart/views/cart.view.result.php* – шаблон подтверждения/опровержения платежа;
- 44. *modules/cart/views/cart.view.show_block.php* – шаблон блока корзины;
- 45. *modules/cart/views/cart.view.table.php* – шаблон таблицы с товарами в корзине;
- 46. *modules/shop/admin/shop.admin.php* – редактирование товаров;
- 47. *modules/shop/admin/shop.admin.additionalcost.php* – дополнительная стоимость;
- 48. *modules/shop/admin/shop.admin.ajax.php* – обработка Ajax-запросов в административной части модуля;
- 49. *modules/shop/admin/shop.admin.category.php* – редактирование категорий магазина;
- 50. *modules/shop/admin/shop.admin.config.php* – настройки модуля;
- 51. *modules/shop/admin/shop.admin.config.js* – настройки модуля, JS-сценарий;
- 52. *modules/shop/admin/shop.admin.counter.php* – статистика просмотров;
- 53. *modules/shop/admin/shop.admin.currency.php* – валюты;
- 54. *modules/shop/admin/shop.admin.delivery.php* – редактирование способов доставки;
- 55. *modules/shop/admin/shop.admin.discount.php* – редактирование скидок;
- 56. *modules/shop/admin/shop.admin.discount.js* – редактирование способов доставки, JS-сценарий;
- 57. *modules/shop/admin/shop.admin.import.php* – импорт;
- 58. *modules/shop/admin/shop.admin.importexport.php* – администрирование импорта/экспорт данных;
- 59. *modules/shop/admin/shop.admin.importexport.category.php* – список описанных файлов;
- 60. *modules/shop/admin/shop.admin.importexport.element.php* – импорт/экспорт данных;
- 61. *modules/shop/admin/shop.admin.importexport.element.js* – импорт/экспорт данных, JS-сценарий;
- 62. *modules/shop/admin/shop.admin.inc.php* – подключение модуля к административной части других модулей;
- 63. *modules/shop/admin/shop.admin.js* – редактирование товаров, JS-сценарий;
- 64. *modules/shop/admin/shop.admin.menu.php* – карта ссылок для модуля «Меню на сайте»;

- 65. *modules/shop/admin/shop.admin.order.php* – редактирование заказов;
- 66. *modules/shop/admin/shop.admin.order.count.php* – количество новых заказов для меню административной панели;
- 67. *modules/shop/admin/shop.admin.order.js* – редактирование заказов, JS-сценарий;
- 68. *modules/shop/admin/shop.admin.ordercount.php* – отчет о продажах;
- 69. *modules/shop/admin/shop.admin.orderparam.php* – конструктор формы оформления заказа;
- 70. *modules/shop/admin/shop.admin.orderparam.js* – конструктор формы оформления заказа, JS-сценарий;
- 71. *modules/shop/admin/shop.admin.orderstatus.php* – статусы заказа;
- 72. *modules/shop/admin/shop.admin.param.php* – редактирование дополнительных характеристик товаров;
- 73. *modules/shop/admin/shop.admin.param.js* – редактирование дополнительных характеристик товаров, JS-сценарий;
- 74. *modules/shop/admin/shop.admin.payhistory.php* – история платежей;
- 75. *modules/shop/admin/shop.admin.payment.php* – редактирование методов оплаты;
- 76. *modules/shop/admin/shop.admin.view.php* – шаблон модуля в административной части;
- 77. *modules/shop/admin/shop.admin.waitlist.php* – список ожиданий;
- 78. *modules/shop/admin/shop.admin.wishlist.php* – список желаний в административной части;
- 79. *modules/shop/inc/shop.inc.order.php* – подключение модуля «Магазин» для работы с заказами;
- 80. *modules/shop/inc/shop.inc.price.php* – подключение модуля «Магазин» для работы с ценами;
- 81. *modules/shop/shop.php* – контроллер модуля «Магазин»;
- 82. *modules/shop/shop.1c.php* – интеграция с системой 1С:Предприятие;
- 83. *modules/shop/shop.ajax.php* – обработка запроса при добавлении товара в корзину;
- 84. *modules/shop/shop.compare.js* – JS-сценарий сравнения товаров;
- 85. *modules/shop/shop.export.php* – экспорт товаров;
- 86. *modules/shop/shop.id.js* – JS-сценарий модуля;
- 87. *modules/shop/shop.inc.php* – подключение модуля «Магазин»;
- 88. *modules/shop/shop.install.php* – установка модуля;
- 89. *modules/shop/shop.js* – JS-сценарий модуля;

90. *modules/shop/shop.model.php* – модель модуля «Магазин»;
91. *modules/shop/shop.search.php* – настройки для поисковой индексации для модуля «Поиск»;
92. *modules/shop/shop.show_search.js* – JS-сценарий формы поиска по товарам;
93. *modules/shop/shop.sitemap.php* – карта ссылок для модуля «Карта сайта»;
94. *modules/shop/shop.yandex.php* – выгрузка в систему Яндекс.Маркет;
95. *modules/shop/views/shop.view.buy_form.php* – шаблон кнопки «Купить», в котором характеристики, влияющие на цену выводятся в виде выпадающего списка;
96. *modules/shop/views/shop.view.buy_form_list.php* – шаблон кнопки «Купить», в котором каждой характеристике, влияющей на цену выведена отдельная кнопка «Купить»;
97. *modules/shop/views/shop.view.compare.php* – шаблон страницы сравнения товаров;
98. *modules/shop/views/shop.view.compared_goods_list.php* – шаблон кнопки «Сравнить выбранные товары»;
99. *modules/shop/views/shop.view.compare_form.php* – шаблон кнопки «Сравнить» для товаров;
100. *modules/shop/views/shop.view.compare_param.php* – шаблон дополнительных характеристик товара на странице сравнения;
101. *modules/shop/views/shop.view.first_page.php* – шаблон первой страницы модуля, если в настройках модуля подключен параметр «Использовать категории»;
102. *modules/shop/views/shop.view.id.php* – шаблон страницы товара;
103. *modules/shop/views/shop.view.list.php* – шаблон списка товаров;
104. *modules/shop/views/shop.view.param.php* – шаблон дополнительных характеристик товара;
105. *modules/shop/views/shop.view.show_add_coupon.php* – шаблон формы активации купона;
106. *modules/shop/views/shop.view.show_block.php* – шаблон блока товаров;
107. *modules/shop/views/shop.view.show_block_order_rel.php* – шаблон блока товаров, которые обычно покупают с текущим товаром;
108. *modules/shop/views/shop.view.show_block_rel.php* – шаблон блока похожих товаров;
109. *modules/shop/views/shop.view.show_search.php* – шаблон форма поиска по товарам;
110. *modules/shop/views/shop.view.sort_block.php* – шаблон блока «Сортировать» с ссылками на направление сортировки;
111. *modules/wishlist/views/wishlist.view.form.php* – шаблон формы редактирования списка желаний;
112. *modules/wishlist/views/wishlist.view.table.php* – шаблон таблицы с товарами в списке

желаний;

113. *modules/wishlist/wishlist.php* – контроллер модуля «Список пожеланий»;

114. *modules/wishlist/wishlist.ajax.php* – обработка запроса при пересчете суммы покупки в списке желаний;

115. *modules/wishlist/wishlist.inc.php* – подключение модуля «Список желаний»;

116. *modules/wishlist/wishlist.js* – JS-сценарий модуля «Список желаний»;

117. *modules/wishlist/wishlist.model.php* – модель модуля Список желаний.

Фотогалерея

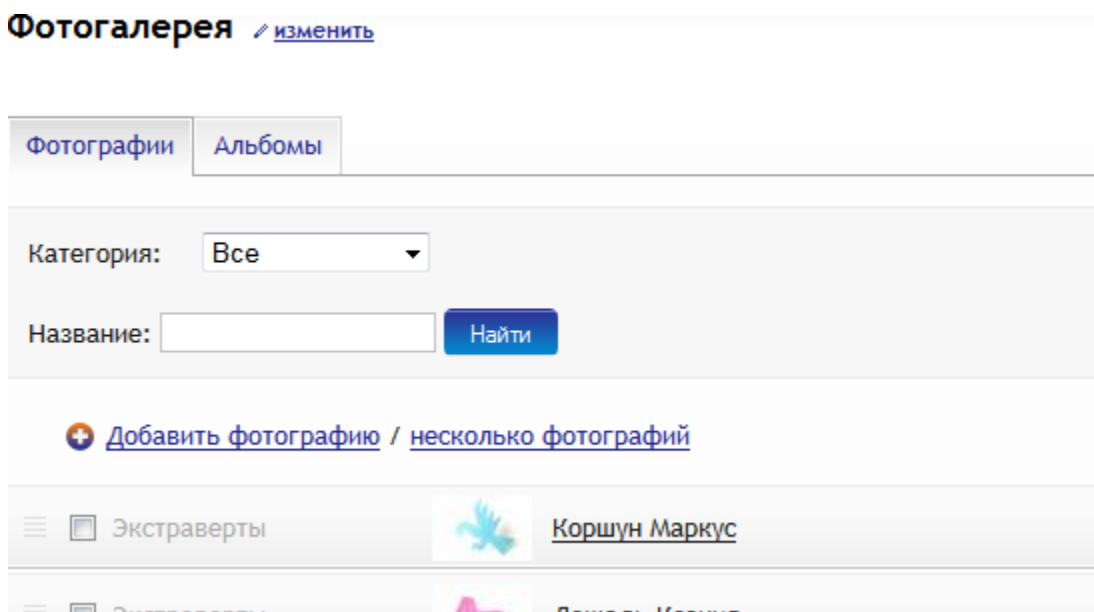
- [Фотографии](#)
- [Альбомы](#)
- [Настройки](#)
- [Шаблонные теги](#)
- [Файлы](#)

Модуль относится к части CMS «Контент» и может быть установлен опционально.

Модуль состоит из двух частей: альбомы и фотографии.

Модуль «Фотогалерея» можно прикреплять к различным страницам сайта. Если модуль прикреплен к нескольким страницам, в административной части модуля появляется специальный фильтр «Раздел сайта» с возможностью фильтрации вывода (списка фотографий, категорий и настроек) по принадлежности к странице.

Фотографии



Порядок отображения фотографий можно менять с помощью мыши, просто перетаскивая их. Если подключены альбомы фотографий, то сортировать фотографии можно только в пределах альбома (в фильтре «Категория» нужно выбрать альбом).

Кнопка ✕ служит для удаления фотографии. Кнопка ⏻ служит для быстрой публикации фотографии (или, наоборот, блокирования).

Редактировать 

Название

Показывать на сайте ☒ ?

Фотография 

Похожие фотографии 

Альбом ?

Фотографии имеют следующие характеристики.

Основные:

- **Название фотографии** – Используется в ссылках на фотографию, заголовках. Параметр имеет разные значения для разных языковых версий сайта.
- **Опубликовать на сайте** – Если не отмечена, фотография не будет отображаться на сайте. Параметр имеет разные значения для разных языковых версий сайта.
- **Фотография** – Фотография будет загружена автоматически после выбора. После загрузки фотография будет обработана автоматически, согласно настройкам модуля.
- **Раздел сайта** – Перенос фотографии на другую страницу сайта, к которой прикреплен модуль. Параметр выводится, если в настройках модуля отключена опция «Использовать категории», если опция подключена, то раздел сайта задается такой же, как у основной категории.
- **Альбом** – Альбом, к которому относится фотография. Возможно выбрать дополнительные альбомы, в которых фотография также будет выводиться. Чтобы выбрать несколько альбомов, удерживайте CTRL. Параметр выводится, если в настройках модуля отмечена опция «Использовать альбомы».
- **Похожие фотографии** – Выбор и добавление к текущей фотографии связей с другими фотографиями. Похожие фотографии выводятся шаблонным тегом `show_block_rel`. По умолчанию связи между фотографиями являются односторонними, это можно изменить, отметив опцию «В блоке похожих фотографий связь двусторонняя» в настройках модуля.
- **Теги** – Добавление тегов к фотографии. Можно добавить либо новый тег, либо открыть и выбрать из уже существующих тегов. Параметр выводится, если в настройках модуля включен параметр «Подключить теги».
- **Счетчик просмотров** – Количество просмотров на сайте текущей фотографии. Статистика ведется и параметр выводится, если в настройках модуля отмечена опция «Подключить счетчик просмотров».
- **Комментарии** – Комментарии, которые оставили пользователи к текущей фотографии. Параметр выводится, если в настройках модуля включен параметр «Показывать комментарии к фотографиям».
- **Рейтинг** – Средний рейтинг, согласно голосованию пользователей сайта. Параметр выводится, если в настройках модуля включен параметр «Подключить рейтинг к фотографиям».

- **Анонс** – Краткое описание фотографии. Если отметить «Добавлять к описанию», на странице элемента анонс выведется вместе с основным описанием. Иначе анонс выведется только в списке, а на отдельной странице будет только описание. Если отметить «Применить типограф», контент будет отформатирован согласно правилам экранной типографики с помощью [веб-сервиса «Типограф»](#). Параметр имеет разные значения для разных языковых версий сайта.
- **Описание** – Полное описание для страницы фотографии. Если отметить «Применить типограф», контент будет отформатирован согласно правилам экранной типографики с помощью [веб-сервиса «Типограф»](#). Параметр имеет разные значения для разных языковых версий сайта.
- **Индексирование для поиска** – Фотография автоматически индексируется для модуля «Поиск по сайту» при внесении изменений.
- **Индексирование для карты сайта** – Фотография автоматически индексируется для карты сайта *sitemap.xml*.

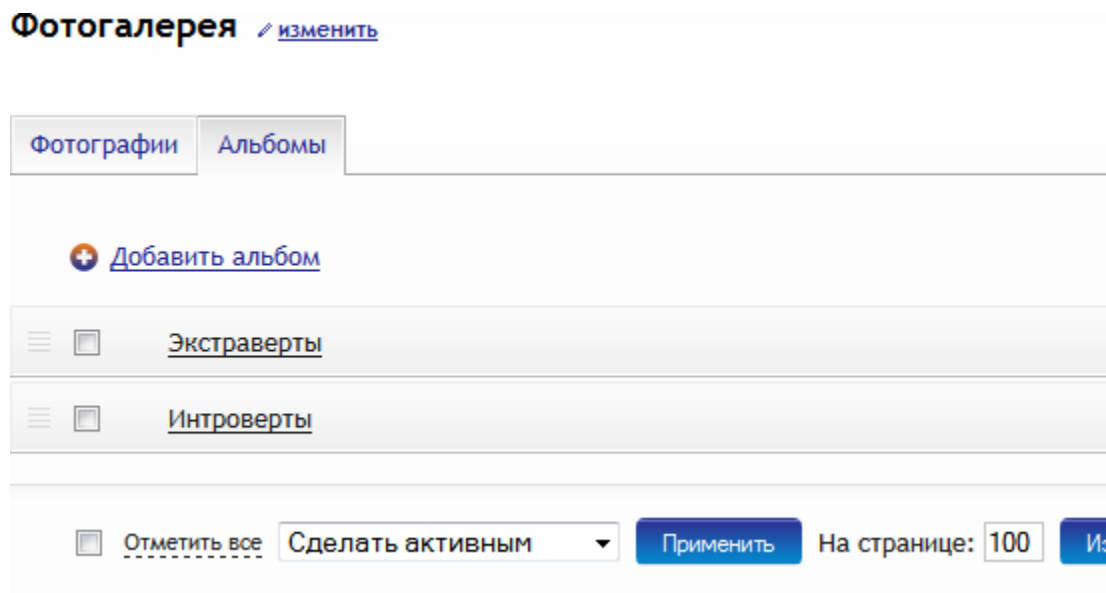
Дополнительные:

- **Номер** – Номер элемента в БД (веб-мастеру и программисту).
- **Редактор** – Изменяется после первого сохранения. Показывает, кто из администраторов сайта первый правил текущую страницу.
- **Время последнего изменения** – Изменяется после сохранения элемента. Отдается в заголовке *Last Modify*.
- **Заголовок окна в браузере, тег Title** – Если не заполнен, тег *Title* будет автоматически сформирован как «Название фотографии – Название страницы – Название сайта», либо согласно шаблонам автоформирования из настроек модуля (SEO-специалисту). Параметр имеет разные значения для разных языковых версий сайта.
- **Ключевые слова, тег Keywords** – Если не заполнен, тег *Keywords* будет автоматически сформирован согласно шаблонам автоформирования из настроек модуля (SEO-специалисту). Параметр имеет разные значения для разных языковых версий сайта.
- **Описание, тег Description** – Если не заполнен, тег *Description* будет автоматически сформирован согласно шаблонам автоформирования из настроек модуля (SEO-специалисту). Параметр имеет разные значения для разных языковых версий сайта.
- **Псевдоссылка** – ЧПУ, т.е. адрес страницы вида: <http://site.ru/pseudossylka/>. Смотрите параметры сайта (SEO-специалисту).
- **Редирект на текущую страницу со страницы** – Позволяет делать редирект с указанной страницы на текущую.
- **Changefreq** – Вероятная частота изменения этой страницы. Это значение используется для генерирования файла *sitemap.xml*. Подробнее читайте в описании [XML-формата файла Sitemap](#) (SEO-специалисту).
- **Priority** – Приоритетность URL относительно других URL на Вашем сайте. Это значение используется для генерирования файла *sitemap.xml*. Подробнее читайте в описании [XML-формата файла Sitemap](#) (SEO-специалисту).
- **Не показывать на карте сайта** – Скрывает отображение ссылки на фотографию в файле *sitemap.xml* и [модуле «Карта сайта»](#).
- **Доступ к текущей фотографии** – Если отметить опцию «Доступ только», фотографию увидят только авторизованные на сайте пользователи, отмеченных типов. Не авторизованные, в том числе поисковые роботы, увидят «404 Страница не найдена» (администратору сайта).
- **Период показа** – Если заполнить, текущая фотография будет опубликована на сайте

в указанный период. В иное время пользователи сайта фотографию не будут видеть, получая ошибку 404 «Страница не найдена» (администратору сайта).

- **Сортировка: установить перед** – Изменить положение текущей фотографии среди других фотографий. Поле доступно для редактирования только для фотографий, отображаемых на сайте (администратору сайта).
- **Шаблон страницы** – Возможность подключить для страницы фотографии шаблон сайта отличный от основного (*themes/site.php*). Все шаблоны для сайта должны храниться в папке *themes* с расширением *.php* (например, *themes/dizain_so_slajdom.php*). Подробнее в [разделе «Шаблоны сайта»](#). (веб-мастеру и программисту, не меняйте этот параметр, если не уверены в результате!).
- **Шаблон модуля** – Шаблон вывода контента модуля на странице отдельной фотографии (веб-мастеру и программисту, не меняйте этот параметр, если не уверены в результате!).

Альбомы



Альбомы фотографий могут иметь неограниченное число вложенных альбомов. При необходимости альбомы можно не использовать. Для этого нужно отключить параметр «Использовать альбомы» в настройках модуля.

Если перед названием альбома имеется знак +, то этому альбому принадлежат вложенные альбомы. Добавить вложенные альбомы можно нажав на знак  рядом с названием альбома.

Порядок отображения альбомов можно менять с помощью мыши, просто перетаскивая их.

Кнопка  служит для удаления альбома. Кнопка  служит для быстрой публикации альбома (или, наоборот, блокирования). При удалении альбома удаляются все вложенные альбомы и фотографии.

Редактировать 

Название

Показывать на сайте ☒

Создать пункт в меню ☐ Меню верхнее
☐ Меню каталог товаров
☐ Меню слева

Комментариев нет

Анонс



Альбом имеют следующие характеристики.

Основные:

- **Название альбома** – Используется в ссылках на альбом, заголовках. Параметр имеет разные значения для разных языковых версий сайта.
- **Опубликовать на сайте** – Если не отмечена, альбом не будет отображаться на сайте. Параметр имеет разные значения для разных языковых версий сайта.
- **Создать пункт в меню** – Если отметить, в [модуле «Меню на сайте»](#) будет создан пункт со ссылкой на текущий альбом.
- **Раздел сайта** – Перенос альбома на другую страницу сайта, к которой прикреплен модуль (администратору сайта).
- **Изображения** – Возможность загрузки нескольких изображений. Варианты размера загружаемых изображений определяются в настройках. Параметр выводится, если в настройках модуля отмечена опция «Использовать изображения для альбомов».
- **Комментарии** – Комментарии, которые оставили пользователи к текущему альбому. Параметр выводится, если в настройках модуля включен параметр «Показывать комментарии к альбомам».
- **Рейтинг** – Средний рейтинг, согласно голосованию пользователей сайта. Параметр выводится, если в настройках модуля включен параметр «Подключить рейтинг к альбомам».
- **Анонс** – Краткое описание альбома. Если отметить «Добавлять к описанию», на странице элемента анонс выведется вместе с основным описанием. Иначе анонс выведется только в списке, а на отдельной странице будет только описание. Если отметить «Применить типограф», контент будет отформатирован согласно правилам экранной типографики с помощью [веб-сервиса «Типограф»](#). Параметр имеет разные значения для разных языковых версий сайта.
- **Описание** – Если отметить «Применить типограф», контент будет отформатирован согласно правилам экранной типографики с помощью [веб-сервиса «Типограф»](#). Параметр имеет разные значения для разных языковых версий сайта.
- **Индексирование для поиска** – Категория автоматически индексируется для модуля «Поиск по сайту» при внесении изменений.
- **Индексирование для карты сайта** – Категория автоматически индексируется для карты сайта *sitemap.xml*.

Дополнительные:

- **Номер** – Номер элемента в БД (веб-мастеру и программисту).
- **Редактор** – Изменяется после первого сохранения. Показывает, кто из администраторов сайта первый правил текущую страницу.
- **Время последнего изменения** – Изменяется после сохранения элемента. Отдается в заголовке *Last Modify*.
- **Заголовок окна в браузере, тег Title** – Если не заполнен, тег *Title* будет автоматически сформирован как «Название категории – Название страницы – Название сайта», либо согласно шаблонам автоформирования из настроек модуля (SEO-специалисту). Параметр имеет разные значения для разных языковых версий сайта.
- **Описание, тег Description** – Если не заполнен, тег *Description* будет автоматически сформирован согласно шаблонам автоформирования из настроек модуля (SEO-специалисту). Параметр имеет разные значения для разных языковых версий сайта.
- **Псевдоссылка** – ЧПУ, т.е. адрес страницы вида: <http://site.ru/pseudossylka/>. Смотрите параметры сайта (SEO-специалисту).
- **Редирект на текущую страницу со страницы** – Позволяет делать редирект с указанной страницы на текущую.
- **Changefreq** – Вероятная частота изменения этой страницы. Это значение используется для генерирования файла *sitemap.xml*. Подробнее читайте в описании [XML-формата файла Sitemap](#) (SEO-специалисту).
- **Priority** – Приоритетность URL относительно других URL на Вашем сайте. Это значение используется для генерирования файла *sitemap.xml*. Подробнее читайте в описании [XML-формата файла Sitemap](#) (SEO-специалисту).
- **Не показывать на карте сайта** – Скрывает отображение ссылки на альбом в файле *sitemap.xml* и [модуле «Карта сайта»](#).
- **Сортировка: установить перед** – Редактирование порядка следования альбома в списке. Поле доступно для редактирования только для альбомов, отображаемых на сайте.
- **Вложенность: принадлежит** – Перемещение текущего альбома и всех его подстраниц в принадлежность другому альбому (администратору сайта).
- **Доступ** – Если отметить опцию «Доступ только», категорию увидят только авторизованные на сайте пользователи, отмеченных типов. Не авторизованные, в том числе поисковые роботы, увидят «404 Страница не найдена» (администратору сайта).
- **Шаблон страницы** – Возможность подключить для страницы альбома шаблон сайта отличный от основного (*themes/site.php*). Все шаблоны для сайта должны храниться в папке *themes* с расширением *.php* (например, *themes/dizain_so_slajdom.php*). Подробнее в [разделе «Шаблоны сайта»](#). (веб-мастеру и программисту, не меняйте этот параметр, если не уверены в результате!).
- **Шаблон модуля** – Шаблон вывода контента модуля на странице списка фотографий в категории (веб-мастеру и программисту, не меняйте этот параметр, если не уверены в результате!).
- **Шаблон модуля для вложенных фотографий** – Шаблон вывода контента модуля на странице отдельной фотографии, вложенной в текущую категорию (веб-мастеру и программисту, не меняйте этот параметр, если не уверены в результате!).

Настройки

Настройки

- **Количество фотографий на странице** – Количество одновременно выводимых фотографий в списке.
- **Использовать альбомы** – Разделение фотогалереи на альбомы-подкатегории.
- **Количество фотографий в списке альбомов** – Количество фотографий, выводимых в списке альбомов на главной странице модуля. Параметр появляется, если отмечена опция «Использовать альбомы».
- **Количество фотографий в списке вложенной категории** – Для первой страницы модуля и для страницы категории. Параметр появляется, если отмечена опция «Использовать альбомы».
- **Показывать фотографии вложенных альбомов** – Если отмечена, в списке фотоальбомов будут отображаться последние добавленные фотографии из всех вложенных альбомов. Параметр появляется, если отмечена опция «Использовать альбомы».
- **Генерировать размеры изображений** – Размеры изображений, заданные в модуле «Изображения» и тег латинскими буквами для подключения изображения на сайте. Обязательно должны быть заданы два размера: превью изображения в списке фотографий (тег medium) и полное изображение (тег large).
- **Использовать изображения для альбомов** – Позволяет включить/отключить загрузку изображений к альбомам.
- **Генерировать размеры изображений для альбомов** – Размеры изображений, заданные в модуле «Изображения» и тег латинскими буквами для подключения изображения на сайте. Обязательно должны быть заданы два размера: превью изображения в списке альбомов (тег medium) и полное изображение (тег large).
- **Отображение изображений в списке альбомов** – Параметр принимает значения:
 - нет (отключает отображение изображений в списке);
 - показывать одно изображение;
 - показывать все изображения. Параметр выводится, если отмечена опция «Использовать изображения».
- **Использовать анимацию при увеличении изображений** – Параметр добавляет JavaScript код, позволяющий включить анимацию при увеличении изображений. Параметр выводится, если отмечена опция «Использовать изображения».
- **Максимальный размер загружаемых файлов** – Параметр показывает максимально

допустимый размер загружаемых файлов, установленный в настройках хостинга. Параметр выводится, если отмечена опция «Использовать изображения».

- **Применить настройки ко всем ранее загруженным изображениям** – Позволяет перекодировать размер уже загруженных изображений. Кнопка необходима, если изменены настройки размеров изображений. Параметр выводится, если отмечена опция «Использовать изображения».
- **Позволяет показывать на сайте фотографии, загруженные с помощью ссылки «Добавить несколько фотографий»..**
- **Открывать фотографию на отдельной странице** – Если не отмечена, фотографии из альбома будут сразу увеличиваться. Если отмечена, каждая фотография будет открываться на отдельной странице с полным текстовым описанием, ее можно будет комментировать, ставить рейтинг.
- **В блоке похожих фотографий связь двусторонняя** – Если отметить, то при назначении фотографии А похожей фотографии Б, у фотографии Б автоматически станет похожей фотография А.
- **Подключить комментарии к фотографиям** – Подключение модуля «Комментарии». Параметр не будет включен, если модуль «Комментарии» не установлен. Подробности см. в разделе [модуль Комментарии](#).
- **Показывать комментарии к альбомам** – Подключение модуля «Комментарии» к альбомам. Параметр не будет включен, если модуль «Комментарии» не установлен. Подробности см. в разделе [модуль Комментарии](#).
- **Подключить теги к фотографиям** – Подключение модуля «Теги». Параметр не будет включен, если модуль «Теги» не установлен. Подробности см. в разделе [модуль «Теги»](#).
- **Подключить рейтинг к фотографиям** – Подключение модуля «Рейтинг». Параметр не будет включен, если модуль «Рейтинг» не установлен. Подробности см. в разделе [модуль «Рейтинг»](#).
- **Подключить рейтинг к категориям** – Подключение модуля «Рейтинг» к альбомам. Параметр не будет включен, если модуль «Рейтинг» не установлен. Подробности см. в разделе [модуль «Рейтинг»](#).
- **Подключить перелинковку** – Отображение перелинковки в модуле. Подробности см. в разделе [модуль «Перелинковка»](#).
- **Подключить счетчик просмотров** – Позволяет считать количество просмотров отдельной фотографии.
- **Выводить счетчик на сайте** – Позволяет вывести на сайте количество просмотров отдельной фотографии. Параметр выводится, если отмечена опция «Счетчик просмотров». Параметр появляется, если отмечена опция «Подключить счетчик просмотров».
- **Шаблон для автоматического генерирования Title** – Если шаблон задан, то заголовок автоматически генерируется по шаблону. В шаблон можно добавить:
 - %title – заданный заголовок,
 - %name – название,
 - %category – название альбома,
 - %parent_category – название альбома верхнего уровня (SEO-специалисту).Параметр имеет разные значения для разных языковых версий сайта.
- **Шаблон для автоматического генерирования Title для альбома** – Если шаблон задан, то заголовок автоматически генерируется по шаблону. В шаблон можно добавить:
 - %title – заданный заголовок,
 - %name – название альбома,

- %parent – название альбома верхнего уровня,
- %page – страница (текст можно поменять в интерфейсе «Языки сайта» – «Перевод интерфейса») (SEO-специалисту).

Параметр имеет разные значения для разных языковых версий сайта. Параметр появляется, если отмечена опция «Использовать альбомы».

- **Шаблон для автоматического генерирования Keywords** – Если шаблон задан, то поле *Keywords* автоматически генерируется по шаблону. В шаблон можно добавить:
 - %keywords – заданные ключевые слова,
 - %name – название,
 - %category – название альбома,
 - %parent_category – название альбома верхнего уровня (SEO-специалисту).

Параметр имеет разные значения для разных языковых версий сайта.

- **Шаблон для автоматического генерирования Keywords для категории** – Если шаблон задан, то поле *Keywords* автоматически генерируется по шаблону. В шаблон можно добавить:
 - %keywords – заданные ключевые слова,
 - %name – название альбома,
 - %parent – название альбома верхнего уровня (SEO-специалисту).

Параметр имеет разные значения для разных языковых версий сайта. Параметр появляется, если отмечена опция «Использовать альбомы».

- **Шаблон для автоматического генерирования Description** – Если шаблон задан, то поле *Description* автоматически генерируется по шаблону. В шаблон можно добавить:
 - %descr – заданное описание,
 - %name – название,
 - %category – название категории,
 - %parent_category – название категории верхнего уровня (SEO-специалисту).

Параметр имеет разные значения для разных языковых версий сайта.

- **Шаблон для автоматического генерирования Description для категории** – Если шаблон задан, то поле *Description* автоматически генерируется по шаблону. В шаблон можно добавить %descr - заданное описание, %name - название категории, %parent - название категории верхнего уровня (SEO-специалисту). Параметр имеет разные значения для разных языковых версий сайта. Параметр появляется, если отмечена опция «Использовать альбомы».
- **Шаблон для списка элементов** – По умолчанию *modules/photo/views/photo.view.list.php*. Параметр для разработчиков! Не устанавливайте, если не уверены в результате.
- **Шаблон для первой страницы модуля (если подключены альбомы)** – По умолчанию *modules/photo/views/photo.view.fitst_page.php*. Параметр для разработчиков! Не устанавливайте, если не уверены в результате.
- **Шаблон для страницы элемента** – По умолчанию, *modules/photo/views/photo.view.id.php*. Параметр для разработчиков! Не устанавливайте, если не уверены в результате.
- **Отдельный пункт в меню администрирования для каждого раздела сайта** – Если модуль подключен к нескольким страницам сайта, отметка данного параметра выведет несколько пунктов в меню административной части для удобства быстрого доступа (администратору сайта).
- **Индексирование для карты сайта** – При изменении настроек, влияющих на отображение страницы, модуль автоматически переиндексируется для карты сайта *sitemap.xml*.

Шаблонные теги

Для работы с модулем «Фотогалерея» служат следующие [шаблонные теги](#):

show_block – выводит несколько фотографий.

Атрибуты:

- **count** – количество выводимых фотографий (по умолчанию 3);
- **site_id** – страницы, к которым прикреплен модуль. Идентификаторы страниц перечисляются через запятую. По умолчанию выбираются все страницы;
- **cat_id** – альбомы фотографий, если в настройках модуля отмечено «Использовать альбомы». Идентификаторы альбомов перечисляются через запятую. По умолчанию альбом не учитывается, выводятся все фотографии;
- **sort** – сортировка фотографий: по умолчанию как на странице модуля, **date** – по дате, **rand** – в случайном порядке;
- **images_variation** – тег размера изображений, задается в настройках модуля;
- **template** – шаблон тега (файл *modules/photo/views/photo.view.show_block_template.php*; по умолчанию шаблон *modules/photo/views/photo.view.show_block.php*).

Пример:

```
<insert name="show_block" module="photo">
```

выведет 3 последние фотографии из фотогалереи

show_block_rel – на странице фотографии выводит похожие фотографии. По умолчанию связи между фотографиями являются односторонними, это можно изменить, отметив опцию «В блоке похожих фотографий связь двусторонняя» в настройках модуля.

Атрибуты:

- **count** – количество выводимых фотографий (по умолчанию 3);
- **images_variation** – тег размера изображений, задается в настройках модуля;
- **template** – шаблон тега (файл *modules/photo/views/photo.view.show_block_rel_template.php*; по умолчанию шаблон *modules/photo/views/photo.view.show_block_rel.php*).

Пример:

```
<insert name="show_block_rel" module="photo">
```


выведет 3 фотографии, прикрепленные к текущей фотографии

Файлы

Модуль находится в папке *modules/photo* и устроен [классически](#).

1. *modules/photo/admin/photo.admin.php* – редактирование фотографий;
2. *modules/photo/admin/photo.admin.category.php* – редактирование категорий фотогалереи;
3. *modules/photo/admin/photo.admin.config.php* – настройки модуля;
4. *modules/photo/admin/photo.admin.config.js* – настройки модуля, JS-сценарий;
5. *modules/photo/admin/photo.admin.counter.php* – статистика просмотров;
6. *modules/photo/admin/photo.admin.js* – редактирование фотографий, JS-сценарий;
7. *modules/photo/admin/photo.admin.menu.php* – карта ссылок для модуля «Меню на сайте»;
8. *modules/photo/photo.php* – контроллер;
9. *modules/photo/photo.install.php* – установка модуля;
10. *modules/photo/photo.js* – JS-сценарий модуля;
11. *modules/photo/photo.model.php* – модель;
12. *modules/photo/photo.search.php* – настройки для поисковой индексации для модуля «Поиск»;
13. *modules/photo/photo.sitemap.php* – карта ссылок для модуля «Карта сайта»;
14. *modules/photo/views/photo.view.first_page.php* – шаблон первой страницы модуля, если в настройках модуля подключен параметр «Использовать категории»;
15. *modules/photo/views/photo.view.id.php* – шаблон страницы фотографии;
16. *modules/photo/views/photo.view.list.php* – шаблон списка фотографий;
17. *modules/photo/views/photo.view.list_tags.php* – шаблон списка фотографий для модуля «Теги»;
18. *modules/photo/views/photo.view.show_block.php* – шаблон блока фотографий;
19. *modules/photo/views/photo.view.show_block_rel.php* – шаблон блока похожих фотографий.

Статьи

- [Статьи](#)
- [Категории](#)
- [Настройки](#)
- [Шаблонные теги](#)
- [Файлы](#)

Модуль «Статьи» отображает статьи.

Модуль относится к части CMS «Контент» и может быть установлен опционально.

Модуль «Статьи» можно прикреплять к различным страницам сайта. Если модуль прикреплен к нескольким страницам, в административной части модуля появляется специальный фильтр «Раздел сайта» с возможностью фильтрации вывода (списка статей, категорий и настроек) по принадлежности к странице.

Модуль состоит из двух частей: категории статей и статьи.

Статьи

Статьи [/ изменить](#)

[Статьи](#) [Категории](#)

Категория:

Название: [Найти](#)

[+ Добавить статью](#)

	Головоногий пингвин-бобёр	Самка слона на свет новс
	Нашествие бобров	Сразу после

В списке статьи выводятся с приоритетом по важности и согласно ручной сортировке.

Кнопка  служит для удаления статьи. Кнопка  служит для быстрой публикации статьи (или, наоборот, блокирования).

Редактировать 

Название

Дата  ?

Показывать на сайте ☒ ?

Изображения 

Статьи имеют следующие характеристики.

Основные:

- **Название** – Используется в ссылках на статью, заголовках. Параметр имеет разные значения для разных языковых версий сайта.
- **Дата** – Вводится в формате дд.мм.гггг чч:мм. Статьи, старше текущей даты начнут отображаться на сайте, начиная с указанной даты.
- **Опубликовать на сайте** – Если не отмечена, статья на сайте не отображается. Параметр имеет разные значения для разных языковых версий сайта.
- **Раздел сайта** – Перенос статьи на другую страницу сайта, к которой прикреплен модуль. Параметр выводится, если в настройках модуля отключена опция «Использовать категории», если опция подключена, то раздел сайта задается такой же, как у основной категории.
- **Категория** – Категория, к которой относится статья. Возможно выбрать дополнительные категории, в которых статья также будет выводиться. Чтобы выбрать несколько категорий, удерживайте CTRL. Параметр выводится, если в настройках модуля отмечена опция «Использовать категории».
- **Изображения** – Изображения будут загружены автоматически после выбора. После загрузки изображения будут обработаны автоматически, согласно настройкам модуля. Параметр выводится, если в настройках модуля отмечена опция «Использовать изображения».
- **Похожие статьи** – Выбор и добавление к текущей статье связей с другими статьями. Похожие статьи выводятся шаблонным тегом `show_block_rel`. По умолчанию связи между статьями являются односторонними, это можно изменить, отметив опцию «В блоке похожих статей связь двусторонняя» в настройках модуля.
- **Теги** – Добавление тегов к статье. Можно добавить либо новый тег, либо открыть и выбрать из уже существующих тегов. Параметр выводится, если в настройках модуля включен параметр «Подключить теги».
- **Счетчик просмотров** – Количество просмотров на сайте текущей статьи. Статистика ведется и параметр выводится, если в настройках модуля отмечена опция «Подключить счетчик просмотров».
- **Комментарии** – Комментарии, которые оставили пользователи к текущей статье. Параметр выводится, если в настройках модуля включен параметр «Показывать комментарии к статьям».

- **Рейтинг** – Средний рейтинг, согласно голосованию пользователей сайта. Параметр выводится, если в настройках модуля включен параметр «Подключить рейтинг к статьям».
- **Анонс** – Краткое описание статьи. Если отметить «Добавлять к описанию», на странице элемента анонс выведется вместе с основным описанием. Иначе анонс выведется только в списке, а на отдельной странице будет только описание. Если отметить «Применить типограф», контент будет отформатирован согласно правилам экранной типографики с помощью [веб-сервиса «Типограф»](#). Параметр имеет разные значения для разных языковых версий сайта.
- **Текст статьи** – Полное текст для страницы статьи. Если отметить «Применить типограф», контент будет отформатирован согласно правилам экранной типографики с помощью [веб-сервиса «Типограф»](#). Параметр имеет разные значения для разных языковых версий сайта.
- **Индексирование для поиска** – Статья автоматически индексируется для модуля «Поиск по сайту» при внесении изменений.
- **Индексирование для карты сайта** – Статья автоматически индексируется для карты сайта *sitemap.xml*.

Дополнительные:

- **Номер** – Номер элемента в БД (веб-мастеру и программисту).
- **Редактор** – Изменяется после первого сохранения. Показывает, кто из администраторов сайта первый правил текущую страницу.
- **Время последнего изменения** – Изменяется после сохранения элемента. Отдается в заголовке *Last Modify*.
- **Заголовок окна в браузере, тег Title** – Если не заполнен, тег *Title* будет автоматически сформирован как «Название статьи – Название страницы – Название сайта», либо согласно шаблонам автоформирования из настроек модуля (SEO-специалисту). Параметр имеет разные значения для разных языковых версий сайта.
- **Ключевые слова, тег Keywords** – Если не заполнен, тег *Keywords* будет автоматически сформирован согласно шаблонам автоформирования из настроек модуля (SEO-специалисту). Параметр имеет разные значения для разных языковых версий сайта.
- **Описание, тег Description** – Если не заполнен, тег *Description* будет автоматически сформирован согласно шаблонам автоформирования из настроек модуля (SEO-специалисту). Параметр имеет разные значения для разных языковых версий сайта.
- **Псевдоссылка** – ЧПУ, т.е. адрес страницы вида: <http://site.ru/pseudossylka/>. Смотрите параметры сайта (SEO-специалисту).
- **Редирект на текущую страницу со страницы** – Позволяет делать редирект с указанной страницы на текущую.
- **Changefreq** – Вероятная частота изменения этой страницы. Это значение используется для генерирования файла *sitemap.xml*. Подробнее читайте в описании [XML-формата файла Sitemap](#) (SEO-специалисту).
- **Priority** – Приоритетность URL относительно других URL на Вашем сайте. Это значение используется для генерирования файла *sitemap.xml*. Подробнее читайте в описании [XML-формата файла Sitemap](#) (SEO-специалисту).
- **Не показывать на карте сайта** – Скрывает отображение ссылки на статью в файле *sitemap.xml* и [модуле «Карта сайта»](#).
- **Доступ** – Если отметить опцию «Доступ только», статью увидят только авторизованные на сайте пользователи, отмеченных типов. Не авторизованные, в том числе поисковые роботы, увидят «404 Страница не найдена» (администратору сайта).

- **Период показа** – Если заполнить, текущая статья будет опубликована на сайте в указанный период. В иное время пользователи сайта статью не будут видеть, получая ошибку 404 «Страница не найдена» (администратору сайта).
- **Сортировка: установить перед** – Изменить положение текущей статьи среди других статей. Поле доступно для редактирования только для статей, отображаемых на сайте (администратору сайта).
- **Шаблон страницы** – Возможность подключить для страницы статьи шаблон сайта отличный от основного (*themes/site.php*). Все шаблоны для сайта должны храниться в папке *themes* с расширением *.php* (например, *themes/dizain_so_slajdom.php*). Подробнее в [разделе «Шаблоны сайта»](#). (веб-мастеру и программисту, не меняйте этот параметр, если не уверены в результате!).
- **Шаблон модуля** – Шаблон вывода контента модуля на странице отдельной статьи (веб-мастеру и программисту, не меняйте этот параметр, если не уверены в результате!).

В модуль «Статьи» встроена автоматическая публикация статей в формате RSS.

Адрес RSS-ленты: <http://site.ru/clauses/rss/>.

У англоязычной версии сайта адрес RSS-ленты новостей будет иметь вид:
<http://site.ru/eng/clauses/rss/>

Категории

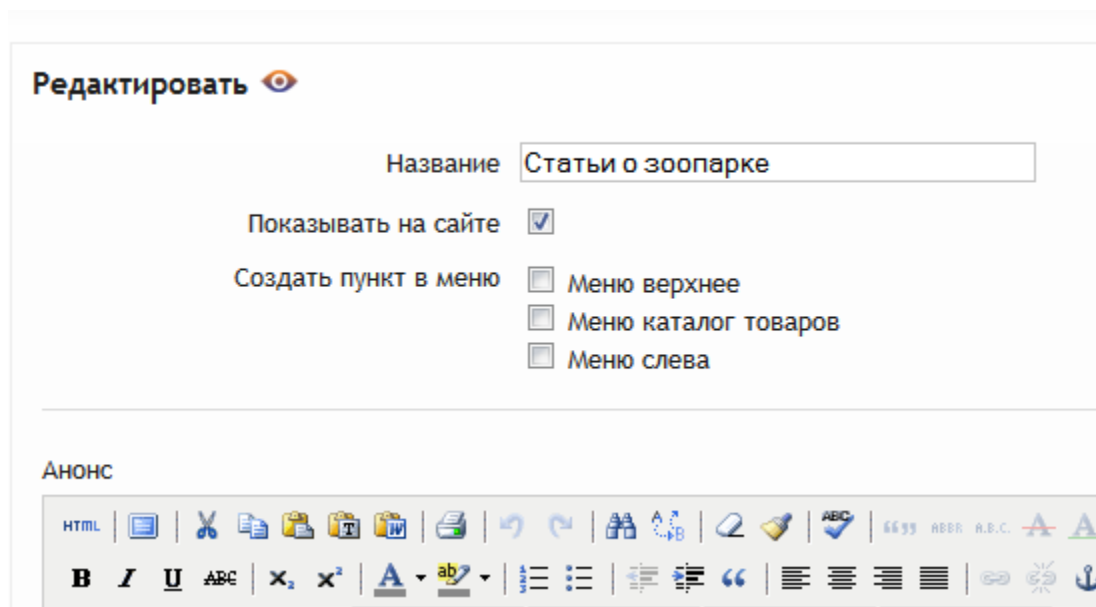
The screenshot shows the 'Статьи' (Articles) management interface. At the top, there's a header with 'Статьи' and a link to 'изменить' (edit). Below this, there are two tabs: 'Статьи' and 'Категории' (Categories), with 'Категории' being the active one. Under the 'Категории' tab, there's a button with a plus icon and the text 'Добавить категорию' (Add category). Below that, there's a search bar with a magnifying glass icon and the text 'Статьи о зоопарке' (Articles about the zoo). At the bottom, there's a filter bar with a checkbox for 'Отметить все' (Mark all), a dropdown menu for 'Сделать активным' (Make active), a blue 'Применить' (Apply) button, and a 'На странице: 100' (Items per page: 100) dropdown.

Категории статей могут иметь неограниченное число подкатегорий. При необходимости категории можно не использовать. Для этого нужно отключить параметр «Использовать категории» в настройках модуля.

Если перед названием категории имеется знак +, то этой категории принадлежат подкатегории. Добавить подкатеорию к категории можно нажав на знак  рядом с названием категории.

Порядок отображения категорий можно редактировать с помощью мыши и кнопки .

Кнопка ✕ служит для удаления категории. Кнопка ⏻ служит для быстрой публикации категории (или, наоборот, блокирования). При удалении категории удаляются все вложенные подкатегории и статьи.



Категории имеют следующие характеристики.

Основные:

- **Название** – Используется в ссылках на категорию, заголовках. Параметр имеет разные значения для разных языковых версий сайта.
- **Опубликовать на сайте** – Если не отмечена, категорию не увидят посетители сайта. Параметр имеет разные значения для разных языковых версий сайта.
- **Создать пункт в меню** – Если отметить, в [модуле «Меню на сайте»](#) будет создан пункт со ссылкой на текущую категорию.
- **Изображения** – Возможность загрузки нескольких изображений. Варианты размера загружаемых изображений определяются в настройках. Параметр выводится, если в настройках модуля отмечена опция «Использовать изображения для категорий».
- **Комментарии** – Комментарии, которые оставили пользователи к текущей категории. Параметр выводится, если в настройках модуля включен параметр «Показывать комментарии к категориям».
- **Рейтинг** – Средний рейтинг, согласно голосованию пользователей сайта. Параметр выводится, если в настройках модуля включен параметр «Подключить рейтинг к категориям».
- **Анонс** – Краткое описание категории. Если отметить «Добавлять к описанию», на странице элемента анонс выведется вместе с основным описанием. Иначе анонс выведется только в списке, а на отдельной странице будет только описание. Если отметить «Применить типограф», контент будет отформатирован согласно правилам экранной типографики с помощью [веб-сервиса «Типограф»](#). Параметр имеет разные значения для разных языковых версий сайта.
- **Описание** – Если отметить «Применить типограф», контент будет отформатирован согласно правилам экранной типографики с помощью [веб-сервиса «Типограф»](#). Параметр имеет разные значения для разных языковых версий сайта.
- **Индексирование для поиска** – Категория автоматически индексируется для модуля «Поиск по сайту» при внесении изменений.

- **Индексирование для карты сайта** – Категория автоматически индексируется для карты сайта *sitemap.xml*.

Дополнительные:

- **Номер** – Номер элемента в БД (веб-мастеру и программисту).
- **Редактор** – Изменяется после первого сохранения. Показывает, кто из администраторов сайта первый правил текущую страницу.
- **Время последнего изменения** – Изменяется после сохранения элемента. Отдается в заголовке *Last Modify*.
- **Заголовок окна в браузере, тег Title** – Если не заполнен, тег *Title* будет автоматически сформирован как «Название категории – Название страницы – Название сайта», либо согласно шаблонам автоформирования из настроек модуля (SEO-специалисту). Параметр имеет разные значения для разных языковых версий сайта.
- **Ключевые слова, тег Keywords** – Если не заполнен, тег *Keywords* будет автоматически сформирован согласно шаблонам автоформирования из настроек модуля (SEO-специалисту). Параметр имеет разные значения для разных языковых версий сайта.
- **Описание, тег Description** – Если не заполнен, тег *Description* будет автоматически сформирован согласно шаблонам автоформирования из настроек модуля (SEO-специалисту). Параметр имеет разные значения для разных языковых версий сайта.
- **Псевдоссылка** – ЧПУ, т.е. адрес страницы вида: <http://site.ru/pseudossylka/>. Смотрите параметры сайта (SEO-специалисту).
- **Редирект на текущую страницу со страницы** – Позволяет делать редирект с указанной страницы на текущую.
- **Changefreq** – Вероятная частота изменения этой страницы. Это значение используется для генерирования файла *sitemap.xml*. Подробнее читайте в описании [XML-формата файла Sitemap](#) (SEO-специалисту).
- **Priority** – Приоритетность URL относительно других URL на Вашем сайте. Это значение используется для генерирования файла *sitemap.xml*. Подробнее читайте в описании [XML-формата файла Sitemap](#) (SEO-специалисту).
- **Не показывать на карте сайта** – Скрывает отображение ссылки на категорию в файле *sitemap.xml* и [модуле «Карта сайта»](#).
- **Раздел сайта** – Перенос категории на другую страницу сайта, к которой прикреплен модуль (администратору сайта).
- **Вложенность: принадлежит** – Перемещение текущей категории и всех её подкатегорий в принадлежность другой категории (администратору сайта).
- **Сортировка: установить перед** – Редактирование порядка следования категории в списке. Поле доступно для редактирования только для категорий, отображаемых на сайте.
- **Доступ** – Если отметить опцию «Доступ только», категорию увидят только авторизованные на сайте пользователи, отмеченных типов. Не авторизованные, в том числе поисковые роботы, увидят «404 Страница не найдена» (администратору сайта).
- **Шаблон страницы** – Возможность подключить для страницы категории шаблон сайта отличный от основного (*themes/site.php*). Все шаблоны для сайта должны храниться в папке *themes* с расширением *.php* (например, *themes/dizain_so_slajdom.php*). Подробнее в [разделе «Шаблоны сайта»](#). (веб-мастеру и программисту, не меняйте этот параметр, если не уверены в результате!).
- **Шаблон модуля** – Шаблон вывода контента модуля на странице списка статей в категории (веб-мастеру и программисту, не меняйте этот параметр, если не уверены в

результате!).

- **Шаблон модуля для вложенных статей** – Шаблон вывода контента модуля на странице отдельной статьи, вложенной в текущую категорию (веб-мастеру и программисту, не меняйте этот параметр, если не уверены в результате!).

Настройки

Настройки

Количество статей на странице

Использовать категории ☒

Количество статей в списке категорий

Количество статей в списке вложенной категории

Показывать статьи подкатегорий ☐

Использовать изображения ☒

- **Количество статей на странице** – Количество одновременно выводимых статей в списке.
- **Использовать категории** – Позволяет включить/отключить категории статей.
- **Количество статей в списке категорий** – Количество статей, выводимых в списке категорий на главной странице модуля. Параметр появляется, если отмечена опция «Использовать категории».
- **Количество статей в списке вложенной категории** – Для первой страницы модуля и для страницы категории. Параметр появляется, если отмечена опция «Использовать категории».
- **Показывать статьи подкатегорий** – Если отмечена, в списке статей категории будут отображаться статьи из всех вложенных категорий. Параметр появляется, если отмечена опция «Использовать категории».
- **Использовать изображения** – Позволяет включить/отключить загрузку изображений к статьям.
- **Генерировать размеры изображений** – Размеры изображений, заданные в модуле «Изображения» и тег латинскими буквами для подключения изображения на сайте. Обязательно должны быть заданы два размера: превью изображения в списке статей (тег medium) и полное изображение (тег large).
- **Отображение изображений в списке** – Параметр принимает значения:
 - нет (отключает отображение изображений в списке);
 - показывать одно изображение;
 - показывать все изображения. Параметр выводится, если отмечена опция «Использовать изображения».
- **Использовать изображения для категорий** – Позволяет включить/отключить загрузку изображений к категориям.
- **Генерировать размеры изображений для категорий** – Размеры изображений,

заданные в модуле «Изображения» и тег латинскими буквами для подключения изображения на сайте. Обязательно должны быть заданы два размера: превью изображения в списке категорий (тег medium) и полное изображение (тег large). Параметр выводится, если отмечена опция «Использовать изображения для категорий».

- **Отображение изображений в списке категорий** – Параметр принимает значения:
 - нет (отключает отображение изображений в списке);
 - показывать одно изображение;
 - показывать все изображения. Параметр выводится, если отмечена опция «Использовать изображения для категорий».
- **Использовать анимацию при увеличении изображений** – Параметр добавляет JavaScript код, позволяющий включить анимацию при увеличении изображений. Параметр выводится, если отмечена опция «Использовать изображения».
- **Максимальный размер загружаемых файлов** – Параметр показывает максимально допустимый размер загружаемых файлов, установленный в настройках хостинга. Параметр выводится, если отмечена опция «Использовать изображения».
- **Применить настройки ко всем ранее загруженным изображениям** – Позволяет перекодировать размер уже загруженных изображений. Кнопка необходима, если изменены настройки размеров изображений. Параметр выводится, если отмечена опция «Использовать изображения».
- **Счетчик просмотров** – Позволяет считать количество просмотров отдельной статьи.
- **Выводить счетчик на сайте** – Позволяет вывести на сайте количество просмотров отдельной статьи. Параметр выводится, если отмечена опция «Счетчик просмотров». Параметр появляется, если отмечена опция «Счетчик просмотров».
- **Формат даты** – Позволяет настроить отображение даты в модуле. Возможные значения:
 - 01.05.2013,
 - 01.05.2013 14:45,
 - 1 мая 2013 г.,
 - 1 мая,
 - 1 мая 2013, понедельник,
 - вчера 15:30,
 - не отображать.
- **Подключить комментарии к статьям** – Подключение модуля «Комментарии». Параметр не будет включен, если модуль «Комментарии» не установлен. Подробности см. в разделе [модуль Комментарии](#).
- **Показывать комментарии к категориям** – Подключение модуля «Комментарии» к категориям статей. Параметр не будет включен, если модуль «Комментарии» не установлен. Подробности см. в разделе [модуль Комментарии](#).
- **Подключить теги** – Подключение модуля «Теги». Параметр не будет включен, если модуль «Теги» не установлен. Подробности см. в разделе [модуль Теги](#).
- **Подключить рейтинг статей** – Подключение модуля «Рейтинг». Параметр не будет включен, если модуль «Рейтинг» не установлен. Подробности см. в разделе [модуль «Рейтинг»](#).
- **Подключить рейтинг к категориям** – Подключение модуля «Рейтинг» к категориям. Параметр не будет включен, если модуль «Рейтинг» не установлен. Подробности см. в разделе [модуль «Рейтинг»](#).
- **Подключить перелинковку** – Отображение перелинковки в модуле. Подробности см. в разделе [модуль «Перелинковка»](#).
- **В блоке похожих статей связь двусторонняя** – Если отметить, то при назначении статье А похожей статьи Б, у статьи Б автоматически станет похожей статья А.

- **Шаблон для автоматического генерирования Title** – Если шаблон задан, то заголовок автоматически генерируется по шаблону. В шаблон можно добавить:
 - %title – заданный заголовок,
 - %name – название,
 - %category – название категории,
 - %parent_category – название категории верхнего уровня (SEO-специалисту).
 Параметр имеет разные значения для разных языковых версий сайта.
- **Шаблон для автоматического генерирования Title для категории** – Если шаблон задан, то заголовок автоматически генерируется по шаблону. В шаблон можно добавить:
 - %title – заданный заголовок,
 - %name – название категории,
 - %parent – название категории верхнего уровня,
 - %page – страница (текст можно поменять в интерфейсе «Языки сайта» – «Перевод интерфейса») (SEO-специалисту).

Параметр имеет разные значения для разных языковых версий сайта. Параметр появляется, если отмечена опция «Использовать категории».

- **Шаблон для автоматического генерирования Keywords** – Если шаблон задан, то поле *Keywords* автоматически генерируется по шаблону. В шаблон можно добавить:
 - %keywords – заданные ключевые слова,
 - %name – название,
 - %category – название категории,
 - %parent_category – название категории верхнего уровня (SEO-специалисту).
 Параметр имеет разные значения для разных языковых версий сайта.
- **Шаблон для автоматического генерирования Keywords для категории** – Если шаблон задан, то поле *Keywords* автоматически генерируется по шаблону. В шаблон можно добавить:
 - %keywords – заданные ключевые слова,
 - %name – название категории,
 - %parent – название категории верхнего уровня (SEO-специалисту).

Параметр имеет разные значения для разных языковых версий сайта. Параметр появляется, если отмечена опция «Использовать категории».

- **Шаблон для автоматического генерирования Description** – Если шаблон задан, то поле *Description* автоматически генерируется по шаблону. В шаблон можно добавить:
 - %descr – заданное описание,
 - %name – название,
 - %category – название категории,
 - %parent_category – название категории верхнего уровня (SEO-специалисту).
 Параметр имеет разные значения для разных языковых версий сайта.
- **Шаблон для автоматического генерирования Description для категории** – Если шаблон задан, то поле *Description* автоматически генерируется по шаблону. В шаблон можно добавить %descr - заданное описание, %name - название категории, %parent - название категории верхнего уровня (SEO-специалисту). Параметр имеет разные значения для разных языковых версий сайта. Параметр появляется, если отмечена опция «Использовать категории».
- **Шаблон для списка элементов** – По умолчанию *modules/courses/views/courses.view.list.php*. Параметр для разработчиков! Не устанавливайте, если не уверены в результате.

- **Шаблон для первой страницы модуля (если подключены категории)** – По умолчанию *modules/clauses/views/clauses.view.fitst_page.php*. Параметр для разработчиков! Не устанавливайте, если не уверены в результате.
- **Шаблон для страницы элемента** – По умолчанию, *modules/clauses/views/clauses.view.id.php*. Параметр для разработчиков! Не устанавливайте, если не уверены в результате.
- **Отдельный пункт в меню администрирования для каждого раздела сайта** – Если модуль подключен к нескольким страницам сайта, отметка данного параметра выведет несколько пунктов в меню административной части для удобства быстрого доступа (администратору сайта).
- **Индексирование для карты сайта** – При изменении настроек, влияющих на отображение страницы, модуль автоматически переиндексируется для карты сайта *sitemap.xml*.

Шаблонные теги

Для работы с модулем «Статьи» служат следующие [шаблонные теги](#):

show_block – выводит последние статьи на всех страницах, кроме страницы статей, когда выводится список тех же статей, что и в функции.

Атрибуты:

- **count** – количество выводимых статей (по умолчанию 3);
- **site_id** – страницы, к которым прикреплен модуль. Идентификаторы страниц перечисляются через запятую. По умолчанию выбираются все страницы;
- **cat_id** – категории статей, если в настройках модуля отмечено «Использовать категории». Идентификаторы категорий перечисляются через запятую. По умолчанию категория не учитывается, выводятся все статьи;
- **images** – количество изображений, прикрепленных к статье;
- **images_variation** – тег размера изображений, задается в настройках модуля;
- **sort** – сортировка статей: по умолчанию как на странице модуля, **date** – по дате, **rand** – в случайном порядке, **keywords** – статьи, похожие по названию для текущей страницы;
- **template** – шаблон тега (файл *modules/clauses/views/clauses.view.show_block_template.php*; по умолчанию шаблон *modules/clauses/views/clauses.view.show_block.php*).

Пример:

```
<insert name="show_block" module="clauses">
```

выведет 3 статьи сайта

```
<insert name="show_block" module="clauses" count="5" sort="date">
```

выведет 5 последних статей сайта

```
<insert name="show_block" module="clauses" count="1" cat_id="35">
```

выведет одну статью сайта, из раздела статей с id=35

show_block_rel – на странице статьи выводит похожие статьи. По умолчанию связи между статьями являются односторонними, это можно изменить, отметив опцию «В блоке похожих статей связь двусторонняя» в настройках модуля.

Атрибуты:

- **count** – количество выводимых статей (по умолчанию 3);
- **images** – количество изображений, прикрепленных к статье;
- **images_variation** – тег размера изображений, задается в настройках модуля;
- **template** – шаблон тега (файл *modules/clauses/views/clauses.view.show_block_rel_template.php*; по умолчанию шаблон *modules/clauses/views/clauses.view.show_block_rel.php*).

Формат даты задается в настройках модуля.

Пример:

```
<insert name="show_block_rel" module="clauses">
```

выведет 3 статьи, прикрепленные к текущей статье

Файлы

Модуль находится в папке *modules/clauses* и устроен [классически](#).

1. *modules/clauses/admin/clauses.admin.php* – редактирование статей;
2. *modules/clauses/admin/clauses.admin.category.php* – редактирование категорий статей;
3. *modules/clauses/admin/clauses.admin.config.php* – настройки модуля;
4. *modules/clauses/admin/clauses.admin.counter.php* – статистика просмотров;
5. *modules/clauses/admin/clauses.admin.menu.php* – карта ссылок для модуля «Меню на сайте»;
6. *modules/clauses/clauses.php* – контроллер;

7. *modules/clauses/clauses.install.php* – установка модуля;
8. *modules/clauses/clauses.model.php* – модель;
9. *modules/clauses/clauses.rss.php* – RSS лента статей;
10. *modules/clauses/clauses.search.php* – настройки для поисковой индексации для модуля «Поиск»;
11. *modules/clauses/clauses.sitemap.php* – карта ссылок для модуля «Карта сайта»;
12. *modules/clauses/views/clauses.view.first_page.php* – шаблон первой страницы модуля, если в настройках модуля подключен параметр «Использовать категории»;
13. *modules/clauses/views/clauses.view.id.php* – шаблон страницы статьи;
14. *modules/clauses/views/clauses.view.list.php* – шаблон списка статей;
15. *modules/clauses/views/clauses.view.show_block.php* – шаблон блока статей;
16. *modules/clauses/views/clauses.view.show_block_rel.php* – шаблон блока похожих статей.

Баннеры

- [Баннеры](#)
- [Категории](#)
- [Настройки](#)
- [Шаблонные теги](#)
- [Файлы](#)

Модуль «Баннеры» отображает баннеры, размещенные на сайте.

Модуль относится к части CMS «Контент» и может быть установлен опционально.

Баннеры

Баннеры

Баннеры

Категории

Название:

 [Добавить баннер](#)

☐ [Баннер сквозной о вакансии](#)

☐ [Реклама на главной](#)

Редактировать

Название

Дата  ?

Показывать на сайте ☒

Всего показов ?

Всего кликов ?

☐ Изображение ☐ Флэш ☒ HTML

Вил баннера

```
<div style="width: 500px; height: 70px; background: #fff; text-align: center; padding: 5px;">
<h1>Требуется сторож! Обращаться по телефону 8-800-100-1000</h1>
</div>
```

Баннеры имеют следующие характеристики.

Основные:

- **Название.**
- **Дата** – Вводится в формате дд.мм.гггг чч:мм. Если указать дату позже текущей даты, то баннер начнет отображаться на сайте, начиная с указанной даты.
- **Опубликовать на сайте** – Параметр имеет разные значения для разных языковых версий сайта.
- **Всего показов** – Статистика прошедших показов баннера.
- **Всего кликов** – Статистика прошедших кликов по баннеру.
- **Вид баннера** – Изображение, флэш, HTML.
- **Ссылка** – В полном формате *http://www.site.ru/*.
- **Открывать в новом окне** – Ссылка на баннер будет открываться в новом окне
- **Период показа** – Время, в течение которого будет показываться баннер.
- **Ограничить количество показов** – Ограничение показа до заданного количества.
- **Осталось показов** – Укажите число, сколько раз должен показываться баннер. С каждым показом цифра в этом поле будет уменьшаться, пока не станет 0 (или пустое поле). Параметр появляется, если отмечена опция «Ограничить количество показов».
- **Ограничить количество показов по кликам** – Ограничение показа до заданного количества.
- **Осталось кликов** – Укажите число, обозначающее, через какое количество кликов скрыть изображение баннера. С каждым кликом цифра в этом поле будет уменьшаться, пока не станет 0 (или пустое поле). Параметр появляется, если отмечена опция «Ограничить количество показов по кликам».
- **Ограничить количество показов посетителю в сутки** – Ограничение показа баннера посетителю.
- **Количество показов посетителю в сутки** – Сколько раз показывать баннер одному пользователю (счетчик сохраняется в сессии). Параметр появляется, если отмечена опция «Ограничить количество показов посетителю в сутки».
- **Раздел сайта** – Выбор раздела, в котором будет виден баннер.
- **Номер** – Номер элемента в БД (веб-мастеру и программисту).
- **Категория.**

Категории

Категории

Баннеры Категории

 [Добавить категорию](#)

☐ [Сквозные баннеры](#)

☐ [Отметить все](#) Сделать активным ▼ Применить На странице: 30 Изменить

Баннеры могут иметь неограниченное количество категорий. При необходимости категории можно не использовать. Для этого нужно отключить параметр «Использовать категории» в конфигурации модуля.

Редактировать

Название

Сквозные баннеры

Показывать на сайте

☒

Номер

id=1 ?

Сохранить

Сохранить и выйти

Сохранить и добавить ещё

В

Категории имеют следующие характеристики.

Основные:

- **Название.**
- **Опубликовать на сайте.**
- **Номер** – Номер элемента в БД (веб-мастеру и программисту).

Настройки

В настройках модуля вы можете выбрать параметр «Использовать категории».

- **Использовать категории** – Позволяет включить/отключить категории баннеров.

Настройка

Использовать категории

☒

Сохранить

Шаблонные теги

Для работы с модулем «Баннеры» служат следующие [шаблонные теги](#):

show_block – выводит баннеры.

Атрибуты:

- **count** – количество выводимых баннеров. По умолчанию 1. Значение **all** выведет все баннеры;
- **id** – идентификатор баннера, если задан, атрибут **count** игнорируется;
- **sort** – сортировка баннеров: **date** – по дате (по умолчанию), **rand** – в случайном порядке;
- **cat_id** – категория баннеров, если в настройках модуля отмечено «Использовать категории»;
- **template** – шаблон тега (файл *modules/bs/views/bs.view.show_block_template.php*; по умолчанию шаблон *modules/bs/views/bs.view.show_block.php*).

Пример:

```
<insert name="show_block" module="bs">
```

выведет 1 случайный баннер

Файлы

1. *modules/bs/admin/bs.admin.php* – редактирование баннеров;
2. *modules/bs/admin/bs.admin.category.php* – редактирование категорий баннеров;
3. *modules/bs/admin/bs.admin.config.php* – настройки модуля;
4. *modules/bs/bs.php* – контроллер;
5. *modules/bs/bs.ajax.php* – обработка запроса при клике на ссылку баннера;
6. *modules/bs/bs.install.php* – установка модуля;
7. *modules/bs/bs.js* – JS-сценарий обработки клика по баннеру;
8. *modules/bs/bs.model.php* – модель;
9. *modules/bs/views/bs.view.show_block.php* – шаблон блока баннеров.

Файловый архив

- [Файлы](#)
- [Категории](#)
- [Настройки](#)
- [Шаблонные теги](#)
- [Файлы](#)

Модуль относится к части CMS «Контент» и может быть установлен опционально.

Модуль состоит из двух частей: категории файлов и файлы.

Модуль «Файловый архив» можно прикреплять к различным страницам сайта. Если модуль прикреплен к нескольким страницам, в административной части модуля появляется специальный фильтр «Раздел сайта» с возможностью фильтрации вывода (списка файлов, категорий и настроек) по принадлежности к странице.

Файлы

Жизнь зоопарка [изменить](#)

[Файлы](#) [Категории](#)

Категория:

Название:

[+ Добавить файл](#)

	 Кони Прежевалинского-Россомахина
	 Россомахина

Порядок отображения файлов можно менять с помощью мыши, просто перетаскивая их. Если подключены категории файлов, то сортировать файлы можно только в пределах категории (в фильтре «Категория» нужно выбрать категорию файлов). Кнопка  служит для удаления файлов. Кнопка  служит для быстрой публикации файлов (или, наоборот, блокирования).

Редактировать 

☐ [koni-prezhevalinskogo-rossomaxina.txt](#)

Файл ☒ Загрузить файл ☐ Загрузить файл по ссылке

Об:

Название

Показывать на сайте ☒

Изображения 

Файлы имеют следующие характеристики.

Основные:

- **Файлы** – Загружаемые файлы. Доступные типы файлов %attachment_extensions изменяются в настройках модуля (параметр «Доступные типы файлов»). Можно указать ссылку для загрузки файла из удаленного источника или ссылку на файл для скачивания.
- **Название** – Используется в ссылках на файл, заголовках. Параметр имеет разные значения для разных языковых версий сайта.
- **Опубликовать на сайте** – Если не отмечена, файл не будет отображаться на сайте. Параметр имеет разные значения для разных языковых версий сайта.
- **Изображения** – Изображения будут загружены автоматически после выбора. После загрузки изображения будут обработаны автоматически, согласно настройкам модуля. Параметр выводится, если в настройках модуля отмечена опция «Использовать изображения».
- **Похожие файлы** – Выбор и добавление к текущему файлу связей с другими файлами. Похожие файлы выводятся шаблонным тегом show_block_rel. По умолчанию связи между файлами являются односторонними, это можно изменить, отметив опцию «В блоке похожих файлов связь двусторонняя» в настройках модуля.
- **Категория** – Категория, к которой относится файл. Возможно выбрать дополнительные категории, в которых файл также будет выводиться. Чтобы выбрать несколько категорий, удерживайте CTRL. Параметр выводится, если в настройках модуля отмечена опция «Использовать категории».
- **Теги** – Добавление тегов к файлу. Можно добавить либо новый тег, либо открыть и выбрать из уже существующих тегов. Параметр выводится, если в настройках модуля включен параметр «Подключить теги».
- **Раздел сайта** – Перенос файла на другую страницу сайта, к которой прикреплен модуль. Параметр выводится, если в настройках модуля отключена опция «Использовать категории», если опция подключена, то раздел сайта задается такой же, как у основной категории.
- **Счетчик просмотров** – Количество просмотров на сайте текущей файла. Статистика ведется и параметр выводится, если в настройках модуля отмечена опция «Подключить счетчик просмотров».
- **Комментарии** – Комментарии, которые оставили пользователи к текущему файлу.

Параметр выводится, если в настройках модуля включен параметр «Показывать комментарии к файлам».

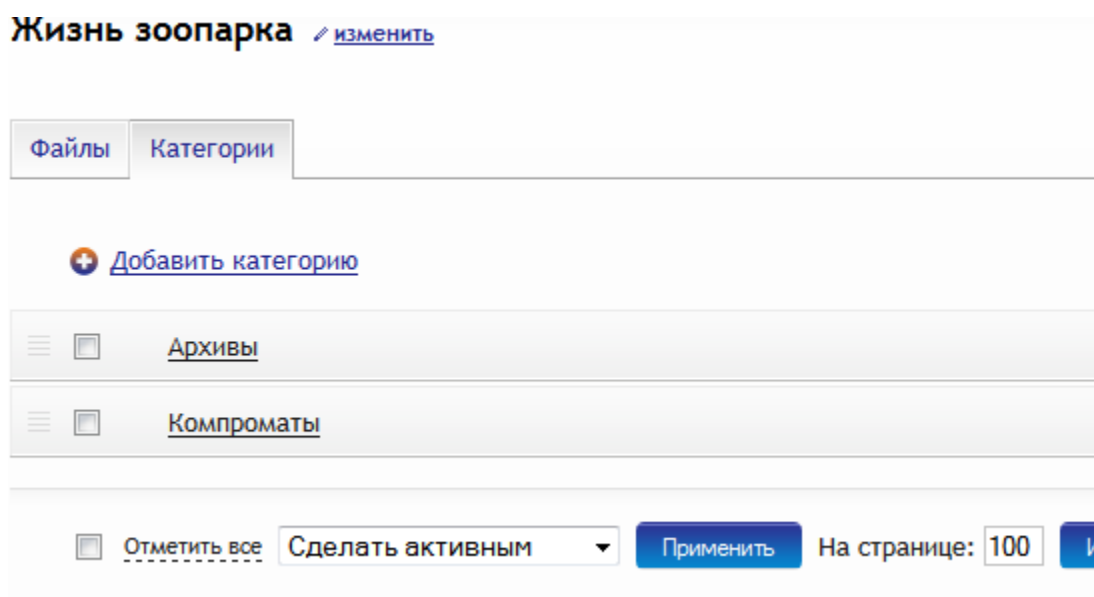
- **Рейтинг** – Средний рейтинг, согласно голосованию пользователей сайта. Параметр выводится, если в настройках модуля включен параметр «Подключить рейтинг к файлам».
- **Анонс** – Краткое описание файла. Выводится в списке файлов и в блоках. Если отметить «Добавлять к описанию», на странице файла анонс выведется вместе с основным описанием. Иначе анонс выведется только в списке, а на отдельной странице будет только описание. Если отметить «Применить типограф», контент будет отформатирован согласно правилам экранной типографики с помощью [веб-сервиса «Типограф»](#). Параметр имеет разные значения для разных языковых версий сайта.
- **Описание** – Полное описание для страницы файла. Если отметить «Применить типограф», контент будет отформатирован согласно правилам экранной типографики с помощью [веб-сервиса «Типограф»](#). Параметр имеет разные значения для разных языковых версий сайта.
- **Индексирование для поиска** – Файл автоматически индексируется для модуля «Поиск по сайту» при внесении изменений.
- **Индексирование для карты сайта** – Файл автоматически индексируется для карты сайта *sitemap.xml*.

Дополнительные:

- **Номер** – Номер элемента в БД (веб-мастеру и программисту).
- **Заголовок окна в браузере, тег Title** – Если не заполнен, тег *Title* будет автоматически сформирован как «Название файла – Название страницы – Название сайта», либо согласно шаблонам автоформирования из настроек модуля (SEO-специалисту). Параметр имеет разные значения для разных языковых версий сайта.
- **Ключевые слова, тег Keywords** – Если не заполнен, тег *Keywords* будет автоматически сформирован согласно шаблонам автоформирования из настроек модуля (SEO-специалисту). Параметр имеет разные значения для разных языковых версий сайта.
- **Описание, тег Description** – Если не заполнен, тег *Description* будет автоматически сформирован согласно шаблонам автоформирования из настроек модуля (SEO-специалисту). Параметр имеет разные значения для разных языковых версий сайта.
- **Псевдоссылка** – ЧПУ (человеко-понятные урл url), адрес страницы вида: *http://site.ru/pseudossylka/*. Смотрите параметры сайта.
- **Редирект на текущую страницу со страницы** – Позволяет делать редирект с указанной страницы на текущую.
- **Changefreq** – Вероятная частота изменения этой страницы. Это значение используется для генерирования файла *sitemap.xml*. Подробнее читайте в описании [XML-формата файла Sitemap](#) (SEO-специалисту).
- **Priority** – Приоритетность URL относительно других URL на Вашем сайте. Это значение используется для генерирования файла *sitemap.xml*. Подробнее читайте в описании [XML-формата файла Sitemap](#) (SEO-специалисту).
- **Доступ** – Если отметить опцию «Доступ только», файл увидят только авторизованные на сайте пользователи, отмеченных типов. Не авторизованные, в том числе поисковые роботы, увидят «404 Страница не найдена» (администратору сайта).
- **Период показа** – Если заполнить, текущий файл будет опубликована на сайте в указанный период. В иное время пользователи сайта файл не будут видеть, получая ошибку 404 «Страница не найдена» (администратору сайта).

- **Сортировка: установить перед** – Изменить положение текущего файла среди других файлов. Поле доступно для редактирования только для файлов, отображаемых на сайте (администратору сайта).
- **Не показывать на карте сайта** – Скрывает отображение ссылки на файл в файле *sitemap.xml* и [модуле «Карта сайта»](#).
- **Шаблон страницы** – Возможность подключить для страницы файла шаблон сайта отличный от основного (*themes/site.php*). Все шаблоны для сайта должны храниться в папке *themes* с расширением *.php* (например, *themes/dizain_so_slajdom.php*). Подробнее в [разделе «Шаблоны сайта»](#). (веб-мастеру и программисту, не меняйте этот параметр, если не уверены в результате!).
- **Шаблон модуля** – Шаблон вывода контента модуля на странице отдельного файла (веб-мастеру и программисту, не меняйте этот параметр, если не уверены в результате!).
- **Редактор** – Изменяется после первого сохранения. Показывает, кто из администраторов сайта первый правил текущую страницу.
- **Время последнего изменения** – Изменяется после сохранения элемента. Отдается в заголовке *Last Modify*.

Категории



Категории файлов могут иметь неограниченное число подкатегорий. При необходимости категории можно не использовать. Для этого нужно отключить параметр «Использовать категории» в настройках модуля.

Если перед названием категории имеется знак +, то этой категории принадлежат подкатегории. Добавить подкатеорию к категории можно нажав на знак  рядом с названием категории.

Порядок отображения категорий можно менять с помощью мыши, просто перетаскивая их.

Кнопка  служит для удаления категории. Кнопка  служит для быстрой публикации категории (или, наоборот, блокирования). При удалении категории удаляются все вложенные подкатегории и файлы.

Редактировать 

Название

Архивы

Показывать на сайте

☒

Создать пункт в меню

☐ Меню верхнее

☐ Меню каталог товаров

☐ Меню слева

Категории имеют следующие характеристики.

Основные:

- **Название** – Используется в ссылках на категорию, заголовках. Параметр имеет разные значения для разных языковых версий сайта.
- **Опубликовать на сайте** – Если не отмечена, категорию не увидят посетители сайта. Параметр имеет разные значения для разных языковых версий сайта.
- **Создать пункт в меню** – Если отметить, в [модуле «Меню на сайте»](#) будет создан пункт со ссылкой на текущую категорию.
- **Изображения** – Возможность загрузки нескольких изображений. Варианты размера загружаемых изображений определяются в настройках. Параметр выводится, если в настройках модуля отмечена опция «Использовать изображения для категорий».
- **Комментарии** – Комментарии, которые оставили пользователи к текущей категории. Параметр выводится, если в настройках модуля включен параметр «Показывать комментарии к категориям».
- **Рейтинг** – Средний рейтинг, согласно голосованию пользователей сайта. Параметр выводится, если в настройках модуля включен параметр «Подключить рейтинг к категориям».
- **Анонс** – Краткое описание категории. Если отметить «Добавлять к описанию», на странице элемента анонс выведется вместе с основным описанием. Иначе анонс выведется только в списке, а на отдельной странице будет только описание. Если отметить «Применить типограф», контент будет отформатирован согласно правилам экранной типографики с помощью [веб-сервиса «Типограф»](#). Параметр имеет разные значения для разных языковых версий сайта.
- **Описание** – Если отметить «Применить типограф», контент будет отформатирован согласно правилам экранной типографики с помощью [веб-сервиса «Типограф»](#). Параметр имеет разные значения для разных языковых версий сайта.
- **Индексирование для поиска** – Категория автоматически индексируется для модуля «Поиск по сайту» при внесении изменений.
- **Индексирование для карты сайта** – Категория автоматически индексируется для карты сайта *sitemap.xml*.

Дополнительные:

- **Номер** – Номер элемента в БД (веб-мастеру и программисту).
- **Заголовок окна в браузере, тег Title** – Если не заполнен, тег *Title* будет автоматически сформирован как «Название категории – Название страницы – Название сайта», либо согласно шаблонам автоформирования из настроек модуля (SEO-специалисту). Параметр имеет разные значения для разных языковых версий сайта.
- **Ключевые слова, тег Keywords** – Если не заполнен, тег *Keywords* будет автоматически сформирован согласно шаблонам автоформирования из настроек модуля (SEO-специалисту). Параметр имеет разные значения для разных языковых версий сайта.
- **Описание, тег Description** – Если не заполнен, тег *Description* будет автоматически сформирован согласно шаблонам автоформирования из настроек модуля (SEO-специалисту). Параметр имеет разные значения для разных языковых версий сайта.
- **Псевдоссылка** – ЧПУ, т.е. адрес страницы вида: <http://site.ru/pseudossylka/>. Смотрите параметры сайта (SEO-специалисту).
- **Редирект на текущую страницу со страницы** – Позволяет делать редирект с указанной страницы на текущую.
- **Changefreq** – Вероятная частота изменения этой страницы. Это значение используется для генерирования файла *sitemap.xml*. Подробнее читайте в описании [XML-формата файла Sitemap](#) (SEO-специалисту).
- **Priority** – Приоритетность URL относительно других URL на Вашем сайте. Это значение используется для генерирования файла *sitemap.xml*. Подробнее читайте в описании [XML-формата файла Sitemap](#) (SEO-специалисту).
- **Раздел сайта** – Перенос категории на другую страницу сайта, к которой прикреплен модуль (администратору сайта).
- **Вложенность: принадлежит** – Перемещение текущей категории и всех её подкатегорий в принадлежность другой категории (администратору сайта).
- **Сортировка: установить перед** – Редактирование порядка следования категории в списке. Поле доступно для редактирования только для категорий, отображаемых на сайте.
- **Не показывать на карте сайта** – Скрывает отображение ссылки на категорию в файле *sitemap.xml* и [модуле «Карта сайта»](#).
- **Доступ** – Если отметить опцию «Доступ только», категорию увидят только авторизованные на сайте пользователи, отмеченных типов. Не авторизованные, в том числе поисковые роботы, увидят «404 Страница не найдена» (администратору сайта).
- **Шаблон страницы** – Возможность подключить для страницы категории шаблон сайта отличный от основного (*themes/site.php*). Все шаблоны для сайта должны храниться в папке *themes* с расширением *.php* (например, *themes/dizain_so_slajdom.php*). Подробнее в [разделе «Шаблоны сайта»](#). (веб-мастеру и программисту, не меняйте этот параметр, если не уверены в результате!).
- **Шаблон модуля** – Шаблон вывода контента модуля на странице списка файлов в категории (веб-мастеру и программисту, не меняйте этот параметр, если не уверены в результате!).
- **Шаблон модуля для вложенных файлов** – Шаблон вывода контента модуля на странице отдельного файла, вложенного в текущую категорию (веб-мастеру и программисту, не меняйте этот параметр, если не уверены в результате!).
- **Редактор** – Изменяется после первого сохранения. Показывает, кто из администраторов сайта первый правил текущую страницу.
- **Время последнего изменения** – Изменяется после сохранения элемента. Отдается в заголовке *Last Modify*.

Настройки

Настройки

Количество файлов на странице	20
Доступные типы файлов (через запятую)	doc, gif, jpg, jpeg, mpg, pdf, png, txt, zip
Использовать категории	☒
Количество файлов в списке категорий	3
Количество файлов в списке вложенной категории	
Показывать файлы подкатегорий	☐

- **Количество файлов на странице** – Количество одновременно выводимых файлов в списке.
- **Доступные типы файлов (через запятую).**

Пример:

doc, gif, jpg, mpg, pdf, png, txt, zip

- **Использовать категории** – Позволяет включить/отключить категории файлов.
- **Количество файлов в списке категорий** – Количество файлов, выводимых в списке категорий на главной странице модуля. Параметр появляется, если отмечена опция «Использовать категории».
- **Количество файлов в списке вложенной категории** – Для первой страницы модуля и для страницы категории. Параметр появляется, если отмечена опция «Использовать категории».
- **Показывать файлы подкатегорий** – Если отмечена, в списке файлов категории будут отображаться файлы из всех вложенных категорий. Параметр появляется, если отмечена опция «Использовать категории».
- **Использовать изображения** – Позволяет включить/отключить загрузку изображений к файлам.
- **Генерировать размеры изображений** – Размеры изображений, заданные в модуле «Изображения» и тег латинскими буквами для подключения изображения на сайте. Обязательно должны быть заданы два размера: превью изображения в списке файлов (тег medium) и полное изображение (тег large).
- **Отображение изображений в списке** – Параметр принимает значения:
 - нет (отключает отображение изображений в списке);
 - показывать одно изображение;
 - показывать все изображения. Параметр выводится, если отмечена опция «Использовать изображения».
- **Использовать изображения для категорий** – Позволяет включить/отключить загрузку изображений к категориям.
- **Генерировать размеры изображений для категорий** – Размеры изображений, заданные в модуле «Изображения» и тег латинскими буквами для подключения

изображения на сайте. Обязательно должны быть заданы два размера: превью изображения в списке категорий (тег medium) и полное изображение (тег large). Параметр выводится, если отмечена опция «Использовать изображения для категорий».

- **Отображение изображений в списке категорий** – Параметр принимает значения:
 - нет (отключает отображение изображений в списке);
 - показывать одно изображение;
 - показывать все изображения. Параметр выводится, если отмечена опция «Использовать изображения для категорий».
- **Использовать анимацию при увеличении изображений** – Параметр добавляет JavaScript код, позволяющий включить анимацию при увеличении изображений. Параметр выводится, если отмечена опция «Использовать изображения».
- **Максимальный размер загружаемых файлов** – Параметр показывает максимально допустимый размер загружаемых файлов, установленный в настройках хостинга. Параметр выводится, если отмечена опция «Использовать изображения».
- **Применить настройки ко всем ранее загруженным изображениям** – Позволяет перекодировать размер уже загруженных изображений. Кнопка необходима, если изменены настройки размеров изображений. Параметр выводится, если отмечена опция «Использовать изображения».
- **Счетчик просмотров** – Позволяет считать количество просмотров отдельного файла.
- **Выводить счетчик на сайте** – Позволяет вывести на сайте количество просмотров отдельного файла. Параметр выводится, если отмечена опция «Счетчик просмотров». Параметр появляется, если отмечена опция «Счетчик просмотров».
- **Показывать комментарии к файлам** – Подключение модуля «Комментарии». Параметр не будет включен, если модуль «Комментарии» не установлен. Подробности см. в разделе [модуль Комментарии](#).
- **Показывать комментарии к категориям** – Подключение модуля «Комментарии» к категориям файлов. Параметр не будет включен, если модуль «Комментарии» не установлен. Подробности см. в разделе [модуль Комментарии](#).
- **Показывать теги файлов** – Подключение модуля «Теги». Параметр не будет включен, если модуль «Теги» не установлен. Подробности см. в разделе [модуль «Теги»](#).
- **Показывать рейтинг файлов** – Подключение модуля «Рейтинг». Параметр не будет включен, если модуль «Рейтинг» не установлен. Подробности см. в разделе [модуль «Рейтинг»](#).
- **Подключить рейтинг к категориям** – Подключение модуля «Рейтинг» к категориям. Параметр не будет включен, если модуль «Рейтинг» не установлен. Подробности см. в разделе [модуль «Рейтинг»](#).
- **Подключить перелинковку** – Отображение перелинковки в модуле. Подробности см. в разделе [модуль «Перелинковка»](#).
- **В блоке похожих файлов связь двусторонняя** – Позволяет установленную в файле связь с другим файлом использовать в двух направлениях.
- **Шаблон для автоматического генерирования Title** – Если шаблон задан, то заголовок автоматически генерируется по шаблону. В шаблон можно добавить:
 - %title – заданный заголовок,
 - %name – название,
 - %category – название категории,
 - %parent_category – название категории верхнего уровня (SEO-специалисту).Параметр имеет разные значения для разных языковых версий сайта.
- **Шаблон для автоматического генерирования Title для категории** – Если шаблон задан, то заголовок автоматически генерируется по шаблону. В шаблон можно

добавить:

- %title – заданный заголовок,
- %name – название категории,
- %parent – название категории верхнего уровня,
- %page – страница (текст можно поменять в интерфейсе «Языки сайта» – «Перевод интерфейса») (SEO-специалисту).

Параметр имеет разные значения для разных языковых версий сайта. Параметр появляется, если отмечена опция «Использовать категории».

- **Шаблон для автоматического генерирования Keywords** – Если шаблон задан, то поле *Keywords* автоматически генерируется по шаблону. В шаблон можно добавить:
 - %keywords – заданные ключевые слова,
 - %name – название,
 - %category – название категории,
 - %parent_category – название категории верхнего уровня (SEO-специалисту).

Параметр имеет разные значения для разных языковых версий сайта.

- **Шаблон для автоматического генерирования Keywords для категории** – Если шаблон задан, то поле *Keywords* автоматически генерируется по шаблону. В шаблон можно добавить:
 - %keywords – заданные ключевые слова,
 - %name – название категории,
 - %parent – название категории верхнего уровня (SEO-специалисту).

Параметр имеет разные значения для разных языковых версий сайта. Параметр появляется, если отмечена опция «Использовать категории».

- **Шаблон для автоматического генерирования Description** – Если шаблон задан, то поле *Description* автоматически генерируется по шаблону. В шаблон можно добавить:
 - %descr – заданное описание,
 - %name – название,
 - %category – название категории,
 - %parent_category – название категории верхнего уровня (SEO-специалисту).

Параметр имеет разные значения для разных языковых версий сайта.

- **Шаблон для автоматического генерирования Description для категории** – Если шаблон задан, то поле *Description* автоматически генерируется по шаблону. В шаблон можно добавить %descr - заданное описание, %name - название категории, %parent - название категории верхнего уровня. Параметр имеет разные значения для разных языковых версий сайта. Параметр появляется, если отмечена опция «Использовать категории».
- **Шаблон для списка элементов** – По умолчанию *modules/files/views/files.view.list.php*. Параметр для разработчиков! Не устанавливайте, если не уверены в результате.
- **Шаблон для первой страницы модуля (если подключены категории)** – По умолчанию *modules/files/views/files.view.fitst_page.php*. Параметр для разработчиков! Не устанавливайте, если не уверены в результате.
- **Шаблон для страницы элемента** – По умолчанию, *modules/files/views/files.view.id.php*. Параметр для разработчиков! Не устанавливайте, если не уверены в результате.
- **Отдельный пункт в меню администрирования для каждого раздела сайта** – Если модуль подключен к нескольким страницам сайта, отметка данного параметра выведет несколько пунктов в меню административной части для удобства быстрого доступа (администратору сайта).

- **Индексирование для карты сайта** – При изменении настроек, влияющих на отображение страницы, модуль автоматически переиндексируется для карты сайта *sitemap.xml*.

Шаблонные теги

Для работы с модулем «Файловый архив» служат следующие [шаблонные теги](#):

show_block – выводит несколько файлов из файлового архива.

Атрибуты:

- **count** – количество выводимых файлов (по умолчанию 3);
- **site_id** – страницы, к которым прикреплен модуль. Идентификаторы страниц перечисляются через запятую. По умолчанию выбираются все страницы;
- **cat_id** – категории файлов, если в настройках модуля отмечено «Использовать категории». Идентификаторы категорий перечисляются через запятую. По умолчанию категория не учитывается, выводятся все файлы;
- **images** – количество изображений, прикрепленных к файлу;
- **images_variation** – тег размера изображений, задается в настройках модуля;
- **sort** – сортировка файлов: по умолчанию как на странице модуля, **rand** – в случайном порядке, **date** – по дате;
- **template** – шаблон тега (файл *modules/files/views/files.view.show_block_template.php*; по умолчанию шаблон *modules/files/views/files.view.show_block.php*).

Пример:

```
<insert name="show_block" module="files">
```

выведет 3 последние файла из файлового архива

show_block_rel – на странице файлы выводит похожие файлы. По умолчанию связи между файлами являются односторонними, это можно изменить, отметив опцию «В блоке похожих файлов связь двусторонняя» в настройках модуля.

Атрибуты:

- **count** – количество выводимых файлов (по умолчанию 3);
- **images** – количество изображений, прикрепленных к файлу;
- **images_variation** – тег размера изображений, задается в настройках модуля;
- **template** – шаблон тега (файл *modules/files/views/files.view.show_block_rel_template.php*; по умолчанию шаблон

modules/files/views/files.view.show_block_rel.php).

Пример:

```
<insert name="show_block_rel" module="files">
```

выведет 3 файла, прикрепленные к текущему файлу

Файлы

Модуль находится в папке *modules/files* и устроен [классически](#).

1. *modules/files/admin/files.admin.php* – редактирование файлов в файловом архиве;
2. *modules/files/admin/files.admin.category.php* – редактирование категорий файлового архива;
3. *modules/files/admin/files.admin.config.php* – настройки модуля;
4. *modules/files/admin/files.admin.counter.php* – статистика просмотров;
5. *modules/files/admin/files.admin.menu.php* – карта ссылок для модуля «Меню на сайте»;
6. *modules/files/files.php* – контроллер;
7. *modules/files/files.install.php* – установка модуля;
8. *modules/files/files.model.php* – модель;
9. *modules/files/files.search.php* – настройки для поисковой индексации для модуля «Поиск»;
10. *modules/files/files.sitemap.php* – карта ссылок для модуля «Карта сайта»;
11. *modules/files/views/files.view.first_page.php* – шаблон первой страницы модуля, если в настройках модуля подключен параметр «Использовать категории»;
12. *modules/files/views/files.view.id.php* – шаблон страницы файла;
13. *modules/files/views/files.view.list.php* – шаблон вывода списка файлов;
14. *modules/files/views/files.view.show_block.php* – шаблон блока файлов;
15. *modules/files/views/files.view.show_block_rel.php* – шаблон блока похожих файлов.

Объявления

- [Объявления](#)
- [Категории](#)
- [Характеристики](#)
- [Настройки](#)
- [Шаблонные теги](#)
- [Файлы](#)

Модуль «Объявления» отображает текстово-графические материалы с сортировкой по датам.

Модуль относится к части CMS «Контент» и может быть установлен опционально.

Модуль «Объявления» можно прикреплять к различным страницам сайта. Если модуль прикреплен к нескольким страницам, в административной части модуля появляется специальный фильтр «Раздел сайта» с возможностью фильтрации вывода (списка объявлений, категорий и настроек) по принадлежности к странице.

Модуль состоит из трёх частей: категории объявлений, объявления и характеристики объявлений.

Объявления

Объявления [изменить](#)

Объявления

Характеристики

Категории

Категория:

Все

Название:

Найти

+

[Добавить объявление](#)

Продажа бумажных животных	11.04.2013 11:57	Суслик	Брал тут в м гарантия. С
Продажа бумажных	11.04.2013 11:57	Модуль не установлен	В боку застр

В списке объявления сортируются по дате.

Кнопка ✕ служит для удаления объявления. Кнопка ⏻ служит для быстрой публикации объявления (или, наоборот, временной деактивации объявления).

Редактировать 

Заголовок

Дата создания  ?

Показывать на сайте ☒ ?

Комментариев нет

Изображения

Цена

Объявление имеет следующие характеристики.

Основные:

- **Заголовок** – Используется в ссылках на объявление, заголовках. Параметр имеет разные значения для разных языковых версий сайта.
- **Дата создания** – Вводится в формате дд.мм.гггг чч:мм. Если указать дату позже текущей даты, то объявление начнет отображаться на сайте, начиная с указанной даты.
- **Опубликовать на сайте** – Если не отмечена, объявление не будет отображаться на сайте. Параметр имеет разные значения для разных языковых версий сайта.
- **Счетчик просмотров** – Количество просмотров на сайте текущего объявления. Статистика ведется и параметр выводится, если в настройках модуля отмечена опция «Подключить счетчик просмотров».
- **Рейтинг** – Средний рейтинг, согласно голосованию пользователей сайта. Параметр выводится, если в настройках модуля включен параметр «Подключить рейтинг к объявлениям».
- **Комментарии** – Комментарии, которые оставили пользователи к текущему объявлению. Параметр выводится, если в настройках модуля включен параметр «Показывать комментарии к объявлениям».
- **Изображения** – Изображения будут загружены автоматически после выбора. После загрузки изображения будут обработаны автоматически, согласно настройкам модуля. Параметр выводится, если в настройках модуля отмечена опция «Использовать изображения».
- **Характеристики** – Группа полей, определенных в части «Характеристики». Параметр имеет разные значения для разных языковых версий сайта.
- **Похожие объявления** – Похожие объявления выводятся шаблонным тегом `show_block_rel`. По умолчанию связи между объявлениями являются односторонними, это можно изменить, отметив опцию «В блоке похожих объявлений связь двусторонняя» в настройках модуля.
- **Категория** – Категория, к которой относится объявление. Возможно выбрать дополнительные категории, в которых объявление также будет выводиться. Чтобы выбрать несколько категорий, удерживайте CTRL. Параметр выводится, если в настройках модуля отмечена опция «Использовать категории».
- **Раздел сайта** – Перенос объявления на другую страницу сайта, к которой прикреплен

модуль. Параметр выводится, если в настройках модуля отключена опция «Использовать категории», если опция подключена, то раздел сайта задается такой же, как у основной категории.

- **Теги** – Добавление тегов к объявлению. Можно добавить либо новый тег, либо открыть и выбрать из уже существующих тегов. Параметр выводится, если в настройках модуля включен параметр «Подключить теги».
- **Пользователь сайта** – Пользователь, создавший объявление в форме на сайте.
- **Анонс** – Краткое описание объявления. Выводится в списках объявлений и в блоках. Если отметить «Добавлять к описанию», на странице объявления анонс выведется вместе с основным описанием. Иначе анонс выведется только в списке, а на отдельной странице будет только описание. Если отметить «Применить типограф», контент будет отформатирован согласно правилам экранной типографики с помощью [веб-сервиса «Типограф»](#). Параметр имеет разные значения для разных языковых версий сайта.
- **Описание** – Полное описание для страницы объявления. Если отметить «Применить типограф», контент будет отформатирован согласно правилам экранной типографики с помощью [веб-сервиса «Типограф»](#). Параметр имеет разные значения для разных языковых версий сайта.
- **Индексирование для поиска** – Объявление автоматически индексируется для модуля «Поиск по сайту» при внесении изменений.
- **Индексирование для карты сайта** – Объявление автоматически индексируется для карты сайта *sitemap.xml*.

Дополнительные:

- **Номер** – Номер элемента в БД (веб-мастеру и программисту).
- **Заголовок окна в браузере, *тег Title*** – Если не заполнен, тег *Title* будет автоматически сформирован как «Название объявления – Название страницы – Название сайта» Параметр имеет разные значения для разных языковых версий сайта.
- **Ключевые слова, *тег Keywords*** – Если не заполнен, тег *Keywords* будет автоматически сформирован согласно шаблонам автоформирования из настроек модуля (SEO-специалисту). Параметр имеет разные значения для разных языковых версий сайта.
- **Описание, *тег Description*** – Если не заполнен, тег *Description* будет автоматически сформирован согласно шаблонам автоформирования из настроек модуля (SEO-специалисту). Параметр имеет разные значения для разных языковых версий сайта.
- **Псевдоссылка** – ЧПУ (человеко-понятные url), адрес страницы вида: <http://site.ru/pseudossylka/>. Смотрите параметры сайта.
- **Редирект на текущую страницу со страницы** – Позволяет делать редирект с указанной страницы на текущую.
- **Changefreq** – Вероятная частота изменения этой страницы. Это значение используется для генерирования файла *sitemap.xml*. Подробнее читайте в описании [XML-формата файла Sitemap](#) (SEO-специалисту).
- **Priority** – Приоритетность URL относительно других URL на Вашем сайте. Это значение используется для генерирования файла *sitemap.xml*. Подробнее читайте в описании [XML-формата файла Sitemap](#) (SEO-специалисту).
- **Период показа** – Если заполнить, текущее объявление будет опубликована на сайте в указанный период. В иное время пользователи сайта объявление не будут видеть, получая ошибку 404 «Страница не найдена» (администратору сайта).
- **Доступ** – Если отметить опцию «Доступ только», объявление увидят только

авторизованные на сайте пользователи, отмеченных типов. Не авторизованные, в том числе поисковые роботы, увидят «404 Страница не найдена» (администратору сайта).

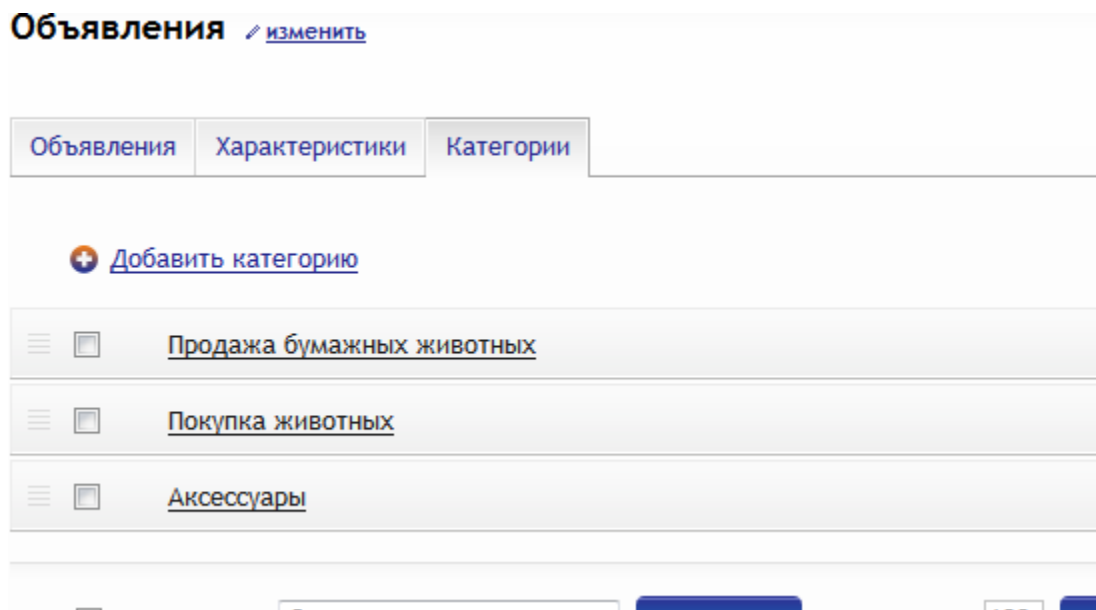
- **Важно (всегда сверху)** – Если отмечена, новость выведется в начале списка, независимо от сортировки по дате. Если важных объявлений несколько, между собой они будут сортироваться по дате.
- **Не показывать на карте сайта** – Скрывает отображение ссылки на объявление в файле *sitemap.xml* и [модуле «Карта сайта»](#).
- **Шаблон страницы** – Возможность подключить для страницы объявления шаблон сайта отличный от основного (*themes/site.php*). Все шаблоны для сайта должны храниться в папке *themes* с расширением *.php* (например, *themes/dizain_so_slajdom.php*). Подробнее в [разделе «Шаблоны сайта»](#). (веб-мастеру и программисту, не меняйте этот параметр, если не уверены в результате!).
- **Шаблон модуля** – Шаблон вывода контента модуля на странице отдельного объявления (веб-мастеру и программисту, не меняйте этот параметр, если не уверены в результате!).
- **Редактор** – Изменяется после первого сохранения. Показывает, кто из администраторов сайта первый правил текущую страницу.
- **Время последнего изменения** – Изменяется после сохранения элемента. Отдается в заголовке *Last Modify*.

В модуль «Объявления» встроена автоматическая публикация объявлений в формате RSS.

Адрес RSS-ленты: <http://site.ru/ab/rss/>.

У англоязычной версии сайта адрес RSS-ленты новостей будет иметь вид:
<http://site.ru/eng/ab/rss/>

Категории



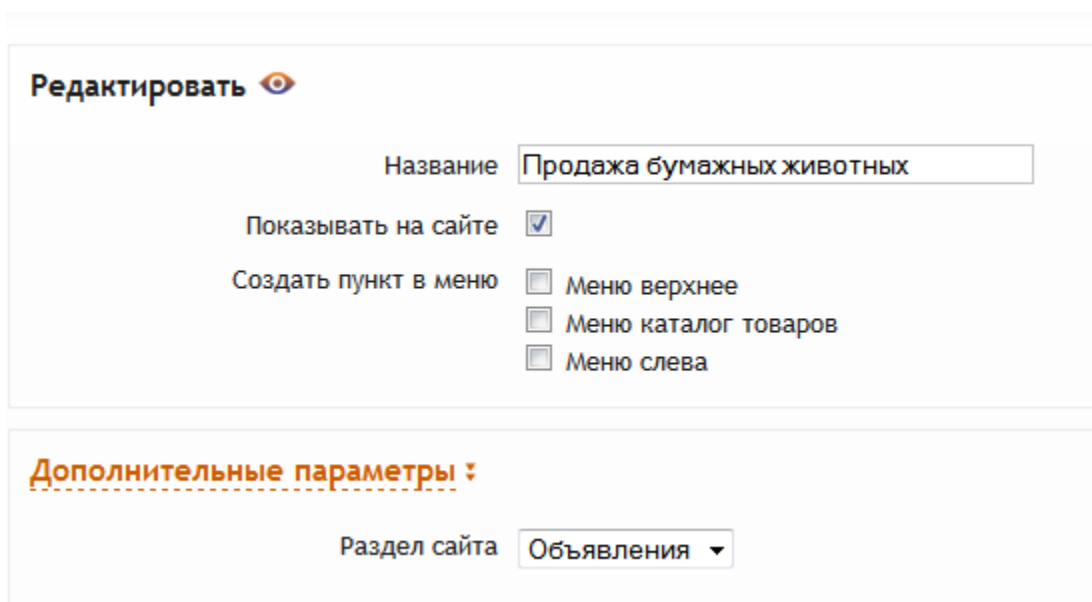
Категории объявлений могут иметь неограниченное число подкатегорий. При необходимости категории можно не использовать. Для этого нужно отключить параметр «Использовать категории» в настройках модуля.

Если перед названием категории имеется знак +, то этой категории принадлежат

подкатегории. Добавить подкатеорию к категории можно нажав на знак  рядом с названием категории.

Порядок отображения категорий можно менять с помощью мыши простым перетаскиванием.

Кнопка  служит для удаления категории. Кнопка  служит для быстрой публикации категории (или, наоборот, временного скрытия на сайте). При удалении категории удаляются все вложенные подкатегории и объявления.



Редактировать 

Название

Показывать на сайте ☒

Создать пункт в меню ☐ Меню верхнее
☐ Меню каталог товаров
☐ Меню слева

Дополнительные параметры 

Раздел сайта

Категории имеют следующие характеристики.

Основные:

- **Название** – Используется в ссылках на категорию, заголовках. Параметр имеет разные значения для разных языковых версий сайта.
- **Опубликовать на сайте** – Если не отмечена, категорию не увидят посетители сайта. Параметр имеет разные значения для разных языковых версий сайта.
- **Создать пункт в меню** – Если отметить, в [модуле «Меню на сайте»](#) будет создан пункт со ссылкой на текущую категорию.
- **Изображения** – Возможность загрузки нескольких изображений. Варианты размера загружаемых изображений определяются в настройках. Параметр выводится, если в настройках модуля отмечена опция «Использовать изображения для категорий».
- **Характеристики** – Список характеристик, применимых к объявлениями текущей категории.
- **Индексирование для поиска** – Категория автоматически индексируется для модуля «Поиск по сайту» при внесении изменений.
- **Индексирование для карты сайта** – Категория автоматически индексируется для карты сайта *sitemap.xml*.

Дополнительные:

- **Раздел сайта** – Перенос категории на другую страницу сайта, к которой прикреплен модуль (администратору сайта).
- **Номер** – Номер элемента в БД (веб-мастеру и программисту).

- **Заголовок окна в браузере, тег Title** – Если не заполнен, тег *Title* будет автоматически сформирован как «Название категории – Название страницы – Название сайта», либо согласно шаблонам автоформирования из настроек модуля (SEO-специалисту). Параметр имеет разные значения для разных языковых версий сайта.
- **Ключевые слова, тег Keywords** – Если не заполнен, тег *Keywords* будет автоматически сформирован согласно шаблонам автоформирования из настроек модуля (SEO-специалисту). Параметр имеет разные значения для разных языковых версий сайта.
- **Описание, тег Description** – Если не заполнен, тег *Description* будет автоматически сформирован согласно шаблонам автоформирования из настроек модуля (SEO-специалисту). Параметр имеет разные значения для разных языковых версий сайта.
- **Псевдоссылка** – ЧПУ, т.е. адрес страницы вида: <http://site.ru/pseudossylka/>. Смотрите параметры сайта (SEO-специалисту).
- **Редирект на текущую страницу со страницы** – Позволяет делать редирект с указанной страницы на текущую.
- **Changefreq** – Вероятная частота изменения этой страницы. Это значение используется для генерирования файла *sitemap.xml*. Подробнее читайте в описании [XML-формата файла Sitemap](#) (SEO-специалисту).
- **Priority** – Приоритетность URL относительно других URL на Вашем сайте. Это значение используется для генерирования файла *sitemap.xml*. Подробнее читайте в описании [XML-формата файла Sitemap](#) (SEO-специалисту).
- **Доступ** – Если отметить опцию «Доступ только», категорию увидят только авторизованные на сайте пользователи, отмеченных типов. Не авторизованные, в том числе поисковые роботы, увидят «404 Страница не найдена» (администратору сайта).
- **Шаблон страницы** – Возможность подключить для страницы категории шаблон сайта отличный от основного (*themes/site.php*). Все шаблоны для сайта должны храниться в папке *themes* с расширением *.php* (например, *themes/dizain_so_slajdom.php*). Подробнее в [разделе «Шаблоны сайта»](#). (веб-мастеру и программисту, не меняйте этот параметр, если не уверены в результате!).
- **Шаблон модуля** – Шаблон вывода контента модуля на странице списка объявлений в категории (веб-мастеру и программисту, не меняйте этот параметр, если не уверены в результате!).
- **Шаблон модуля для вложенных объявлений** – Шаблон вывода контента модуля на странице отдельного объявления, вложенного в текущую категорию (веб-мастеру и программисту, не меняйте этот параметр, если не уверены в результате!).
- **Сортировка: установить перед** – Редактирование порядка следования категории в списке. Поле доступно для редактирования только для категорий, отображаемых на сайте.
- **Не показывать на карте сайта** – Скрывает отображение ссылки на категорию в файле *sitemap.xml* и [модуле «Карта сайта»](#).
- **Вложенность: принадлежит** – Перемещение текущей категории и всех её подкатегорий в принадлежность другой категории (администратору сайта).
- **Комментарии** – Комментарии, которые оставили пользователи к текущей категории. Параметр выводится, если в настройках модуля включен параметр «Показывать комментарии к категориям».
- **Рейтинг** – Средний рейтинг, согласно голосованию пользователей сайта. Параметр выводится, если в настройках модуля включен параметр «Подключить рейтинг к категориям».
- **Анонс** – Краткое описание категории. Если отметить «Добавлять к описанию», на странице элемента анонс выведется вместе с основным описанием. Иначе анонс

выведется только в списке, а на отдельной странице будет только описание. Если отметить «Применить типограф», контент будет отформатирован согласно правилам экранной типографики с помощью [веб-сервиса «Типограф»](#). Параметр имеет разные значения для разных языковых версий сайта.

- **Описание** – Если отметить «Применить типограф», контент будет отформатирован согласно правилам экранной типографики с помощью [веб-сервиса «Типограф»](#). Параметр имеет разные значения для разных языковых версий сайта.
- **Редактор** – Изменяется после первого сохранения. Показывает, кто из администраторов сайта первый правил текущую страницу.
- **Время последнего изменения** – Изменяется после сохранения элемента. Отдается в заголовке *Last Modify*.

Характеристики

Характеристики

Объявления

Характеристики

Категории

Категория:

Все

 [Добавить характеристику](#)

 Цена	число
 Фотография	изображения
 Продавец	заголовок группы характеристик

Порядок отображения характеристик объявлений можно редактировать с помощью мыши и кнопки . Кнопка  служит для удаления характеристики.

Характеристики – это дополнительные параметры, характеризующие объявление. Можно добавлять характеристики общие (для всех объявлений в пределах одного раздела сайта) или присваивать характеристику одной или нескольким категориям объявлений.

Редактировать

Название ?

Тип ▼

Обязательно для заполнения из пользовательской части ☐

Единица измерения

Категория

Продажа бумажных животных
 Покупка животных
 Аксессуары

Сортировка: установить перед [Цена](#)

Характеристики имеют следующие параметры.

Основные:

- **Название** – Имя дополнительной характеристики объявления, доступно для заполнения при редактировании товара. Параметр имеет разные значения для разных языковых версий сайта.
- **Тип** – Возможные значения:
 - строка,
 - число,
 - дата,
 - дата и время,
 - текстовое поле,
 - галочка,
 - выпадающий список,
 - список с выбором нескольких значений,
 - электронный ящик,
 - поле с визуальным редактором,
 - заголовок группы характеристик,
 - файлы,
 - изображения.
- **Максимальное количество добавляемых файлов** – Количество добавляемых файлов. Если значение равно нулю, то форма добавления файлов не выводится. Параметр выводится, если тип характеристики задан как «файлы».
- **Доступные типы файлов (через запятую)** – Параметр выводится, если тип характеристики задан как «файлы».

Пример:

doc, gif, jpg, mpg, pdf, png, txt, zip

- **Распознавать изображения** – Позволяет прикрепленные файлы в формате JPEG, GIF, PNG отображать как изображения. Параметр выводится, если тип характеристики задан как «файлы».
- **Размер для большого изображения** – Размер изображения, отображаемый в

пользовательской части сайта при увеличении изображения предпросмотра. Параметр выводится, если тип характеристики задан как «файлы» и отмечена опция «Распознавать изображения».

- **Размер для маленького изображения** – Размер изображения предпросмотра. Параметр выводится, если тип характеристики задан как «файлы» и отмечена опция «Распознавать изображения».
- **Использовать анимацию при увеличении изображений** – Параметр добавляет JavaScript код, позволяющий включить анимацию при увеличении изображений. Параметр выводится, если отмечена опция «Распознавать изображения». Параметр выводится, если тип характеристики задан как «файлы» и отмечена опция «Распознавать изображения».
- **Максимальный размер загружаемых файлов** – Параметр показывает максимально допустимый размер загружаемых файлов, установленный в настройках хостинга. Параметр выводится, если тип характеристики задан как «файлы».
- **Генерировать размеры изображений** – Размеры изображений, заданные в модуле «Изображения». Параметр выводится, если тип характеристики задан как «изображение».
- **Значения, псевдоссылка** – Поле появляется для характеристик с типом «галочка», «выпадающий список» и «список с выбором нескольких значений».
- **Обязательно для заполнения из пользовательской части.**
- **Единица измерения** – Параметр выводится, если тип характеристики задан как «число». Параметр имеет разные значения для разных языковых версий сайта.
- **Отдельная страница для значений** – Поле появляется для характеристик с типом «выпадающий список» и «список с выбором нескольких значений». Если опция отмечена, то на сайте появляется страница с объявлениями, объединенными одной характеристикой. Например, город: Краснодар. Странице можно задать ЧПУ, для этого рядом с полем «Значение» выводится дополнительное поле «Псевдоссылка».
- **Категория** – Категории объявлений, для которых действует данный параметр. Если не указана ни одна категория, то характеристика считается общей.
- **Сортировка: установить перед** – Редактирование порядка следования характеристики в списке
- **Использовать в форме поиска** – Позволяет отображать характеристику в форме поиска, выводимой тегом show_search.
- **Показывать в списке** – Позволяет выводить значение характеристики для объявлений в списке объявлений на сайте.
- **Показывать в блоке объявлений** – Позволяет выводить значение характеристики для объявлений в блоке объявлений, выводимом тегом show_block.
- **Отображать параметры в блоке для сортировки объявлений** – Позволяет выводить характеристику в виде ссылки для сортировки объявлений по значению характеристики.
- **Описание характеристики** – Поле используется для комментирования характеристик в списке. В пользовательской части поле не используется. Параметр имеет разные значения для разных языковых версий сайта.

Если выбран тип «галочка», появится 2 дополнительных поля: «да» и «нет». Они позволяют задать отображение параметра в пользовательской части сайта. Если значения пустые, то при показе объявления будет, отображается только название параметра (при активной галочке) и не отображается ничего (при неактивной галочке).

Пример:

Без посредников

Если значения заданы, то отображаются эти значения.

Пример:

Посредник: да

Посредник: нет.

Если выбран тип «выпадающий список» или «список с выбором нескольких значений», то появятся дополнительные поля со значениями. Добавить значение можно, нажав  внизу последнего поля, сортировать значения можно с помощью мыши и кнопки . Кнопка  служит для удаления значений.

Если выбран тип «поле с визуальным редактором», то при редактировании поля будет отображаться визуальный редактор, что позволяет редактировать стиль текста.

Настройки

Настройки

Основные настройки

Количество объявлений на странице

10

Формат даты

1.05.2009

Использовать категории

☒

Количество объявлений в списке категорий

3

Количество объявлений в списке вложенной категории

Основные настройки

- **Количество объявлений на странице** – Количество одновременно выводимых объявлений в списке.
- **Формат даты** – Позволяет настроить отображение даты в модуле. Возможные значения:
 - 01.05.2013,
 - 01.05.2013 14:45,
 - 1 мая 2013 г.,
 - 1 мая,
 - 1 мая 2013, понедельник,
 - вчера 15:30,
 - не отображать.

- **Использовать категории** – Позволяет включить/отключить категории объявлений.
- **Количество объявлений в списке категорий** – Для первой страницы модуля, где выходят по несколько объявлений из всех категорий. Параметр появляется, если отмечена опция «Использовать категории».
- **Количество объявлений в списке вложенной категории** – Для первой страницы модуля и для страницы категории. Параметр появляется, если отмечена опция «Использовать категории».
- **Показывать объявления подкатегорий** – Если отмечена, в списке объявлений категории будут отображаться объявления из всех вложенных категорий. Параметр появляется, если отмечена опция «Использовать категории».
- **Искать по описанию** – Параметр позволяет выводить в модуле поиска по объявлениям поиск по описанию.
- **Искать по названию** – Параметр позволяет выводить в модуле поиска по объявлениям поиск по названию.
- **Счетчик просмотров** – Позволяет считать количество просмотров отдельного объявления.
- **Выводить счетчик на сайте** – Позволяет вывести на сайте количество просмотров отдельного объявления. Параметр выводится, если отмечена опция «Счетчик просмотров». Параметр появляется, если отмечена опция «Счетчик просмотров».
- **Подключить комментарии к объявлениям** – Подключение модуля «Комментарии». Параметр не будет включен, если модуль «Комментарии» не установлен. Подробности см. в разделе [модуль Комментарии](#).
- **Показывать комментарии к категориям** – Подключение модуля «Комментарии» к категориям объявлений. Параметр не будет включен, если модуль «Комментарии» не установлен. Подробности см. в разделе [модуль Комментарии](#).
- **Подключить теги** – Подключение модуля «Теги». Параметр не будет включен, если модуль «Теги» не установлен. Подробности см. в разделе [модуль Теги](#).
- **Показывать рейтинг объявлений** – Подключение модуля «Рейтинг». Параметр не будет включен, если модуль «Рейтинг» не установлен. Подробности см. в разделе [модуль Рейтинг](#).
- **Подключить рейтинг к категориям** – Подключение модуля «Рейтинг» к категориям. Параметр не будет включен, если модуль «Рейтинг» не установлен. Подробности см. в разделе [модуль Рейтинг](#).
- **Подключить перелинковку** – Отображение перелинковки в модуле. Подробности см. в разделе [модуль Перелинковка](#).
- **В блоке похожих объявлений связь двусторонняя** – Позволяет установленную в объявлении связь с другим объявлением использовать в двух направлениях.
- **Шаблон для автоматического генерирования Title** – Если шаблон задан, то заголовок автоматически генерируется по шаблону. В шаблон можно добавить:
 - %title – заданный заголовок,
 - %name – название,
 - %category – название категории,
 - %parent_category – название категории верхнего уровня (SEO-специалисту).
 Параметр имеет разные значения для разных языковых версий сайта.
- **Шаблон для автоматического генерирования Title для категории** – Если шаблон задан, то заголовок автоматически генерируется по шаблону. В шаблон можно добавить:
 - %title – заданный заголовок,
 - %name – название категории,
 - %parent – название категории верхнего уровня,

- %page – страница (текст можно поменять в интерфейсе «Языки сайта» – «Перевод интерфейса») (SEO-специалисту).

Параметр имеет разные значения для разных языковых версий сайта. Параметр появляется, если отмечена опция «Использовать категории».

- **Шаблон для автоматического генерирования Keywords** – Если шаблон задан, то поле *Keywords* автоматически генерируется по шаблону. В шаблон можно добавить:
 - %keywords – заданные ключевые слова,
 - %name – название,
 - %category – название категории,
 - %parent_category – название категории верхнего уровня (SEO-специалисту).

Параметр имеет разные значения для разных языковых версий сайта.

- **Шаблон для автоматического генерирования Keywords для категории** – Если шаблон задан, то поле *Keywords* автоматически генерируется по шаблону. В шаблон можно добавить:
 - %keywords – заданные ключевые слова,
 - %name – название категории,
 - %parent – название категории верхнего уровня (SEO-специалисту).

Параметр имеет разные значения для разных языковых версий сайта. Параметр появляется, если отмечена опция «Использовать категории».

- **Шаблон для автоматического генерирования Description** – Если шаблон задан, то поле *Description* автоматически генерируется по шаблону. В шаблон можно добавить:
 - %descr – заданное описание,
 - %name – название,
 - %category – название категории,
 - %parent_category – название категории верхнего уровня (SEO-специалисту).

Параметр имеет разные значения для разных языковых версий сайта.

- **Шаблон для автоматического генерирования Description для категории** – Если шаблон задан, то поле *Description* автоматически генерируется по шаблону. В шаблон можно добавить %descr - заданное описание, %name - название категории, %parent - название категории верхнего уровня. Параметр имеет разные значения для разных языковых версий сайта. Параметр появляется, если отмечена опция «Использовать категории».
- **Шаблон для списка элементов** – По умолчанию *modules/ab/views/ab.view.list.php*. Параметр для разработчиков! Не устанавливайте, если не уверены в результате.
- **Шаблон для первой страницы модуля (если подключены категории)** – По умолчанию *modules/ab/views/ab.view.fitst_page.php*. Параметр для разработчиков! Не устанавливайте, если не уверены в результате.
- **Шаблон для страницы элемента** – По умолчанию, *modules/ab/views/ab.view.id.php*. Параметр для разработчиков! Не устанавливайте, если не уверены в результате.
- **Шаблон для списка элементов с одинаковой характеристикой** – Параметр для разработчиков! Не устанавливайте, если не уверены в результате.
- **Шаблон для поиска элементов** – Параметр для разработчиков! Не устанавливайте, если не уверены в результате.
- **Шаблон для объявлений пользователя** – Параметр для разработчиков! Не устанавливайте, если не уверены в результате.
- **Отдельный пункт в меню администрирования для каждого раздела сайта** – Если модуль подключен к нескольким страницам сайта, отметка данного параметра выведет несколько пунктов в меню административной части для удобства быстрого доступа (администратору сайта).
- **Индексирование для карты сайта** – При изменении настроек, влияющих на

отображение страницы, модуль автоматически переиндексируется для карты сайта *sitemap.xml*.

Добавление объявления на сайте

- **Добавлять объявления могут только зарегистрированные пользователи.**
- **Использовать защитный код (капчу)** – Для добавления объявления пользователь должен ввести защитный код.
- **Сообщение после отправки** – Сообщение, получаемое пользователем при удачной загрузке объявления из формы на сайте, допускаются HTML-теги для оформления сообщения. Параметр имеет разные значения для разных языковых версий сайта.

Пример:

```
<div align="center"><b>Спасибо, объявление успешно добавлено!</b></div>
```

- **Предмодерация объявлений** – Объявление добавляется из формы на сайте не активным и активируется в административной части сайта.
- **Поля формы** – Позволяет выводить/скрыть обязательные поля (название, краткий анонс, описание объявления, период показа, изображения) в форме добавления объявления.

Изображения

- **Использовать изображения** – Позволяет включить/отключить загрузку изображений к объявлениям.
- **Генерировать размеры изображений** – Размеры изображений, заданные в модуле «Изображения» и тег латинскими буквами для подключения изображения на сайте. Обязательно должны быть заданы два размера: превью изображения в списке объявлений (тег medium) и полное изображение (тег large).
- **Отображение изображений в списке** – Параметр принимает значения:
 - нет (отключает отображение изображений в списке);
 - показывать одно изображение;
 - показывать все изображения. Параметр выводится, если отмечена опция «Использовать изображения».
- **Использовать изображения для категорий** – Позволяет включить/отключить загрузку изображений к категориям.
- **Генерировать размеры изображений для категорий** – Размеры изображений, заданные в модуле «Изображения» и тег латинскими буквами для подключения изображения на сайте. Обязательно должны быть заданы два размера: превью изображения в списке категорий (тег medium) и полное изображение (тег large). Параметр выводится, если отмечена опция «Использовать изображения для категорий».
- **Отображение изображений в списке категорий** – Параметр принимает значения:
 - нет (отключает отображение изображений в списке);
 - показывать одно изображение;
 - показывать все изображения. Параметр выводится, если отмечена опция «Использовать изображения для категорий».
- **Использовать анимацию при увеличении изображений** – Параметр добавляет JavaScript код, позволяющий включить анимацию при увеличении изображений.

Параметр выводится, если отмечена опция «Использовать изображения».

- **Максимальный размер загружаемых файлов** – Параметр показывает максимально допустимый размер загружаемых файлов, установленный в настройках хостинга. Параметр выводится, если отмечена опция «Использовать изображения».
- **Применить настройки ко всем ранее загруженным изображениям** – Позволяет переконвертировать размер уже загруженных изображений. Кнопка необходима, если изменены настройки размеров изображений. Параметр выводится, если отмечена опция «Использовать изображения».

Сообщения и уведомления

- **E-mail, указываемый в обратном адресе пользователю** – Возможные значения:
 - e-mail, указанный в параметрах сайта;
 - другой (при выборе этого значения появляется дополнительное поле **впишите e-mail**).
- **Уведомлять о поступлении новых объявлений на e-mail** – Возможность уведомления администратора о поступлении новых объявлений из формы в пользовательской части сайта.
- **E-mail для уведомлений администратора** – Возможные значения:
 - e-mail, указанный в параметрах сайта;
 - другой (при выборе этого значения появляется дополнительное поле **впишите e-mail**).

Параметр появляется, если отмечена опция «Уведомлять о поступлении новых объявлений на e-mail».

- **Тема письма для уведомлений** – Можно добавлять:
 - %title – название сайта,
 - %url – адрес сайта (например, site.ru).

Параметр появляется, если отмечена опция «Уведомлять о поступлении новых объявлений на e-mail».

- **Сообщение для уведомлений** – Можно добавлять:
 - %title – название сайта,
 - %url – адрес сайта (например, site.ru),
 - %message – объявление.

Параметр появляется, если отмечена опция «Уведомлять о поступлении новых объявлений на e-mail».

- **Уведомлять о поступлении новых объявлений по SMS** – Возможность отправлять SMS администратору при поступлении объявления. Параметр можно подключить, если в [Параметрах сайта](#) настроены SMS-уведомления.
- **Номер телефона в федеральном формате** – Номер телефона для SMS-уведомлений администратора о новом объявлении. Параметр появляется, если отмечена опция «Уведомлять о поступлении новых объявлений по SMS».
- **Сообщение для уведомлений** – Текст сообщения для SMS-уведомлений администратора о новом объявлении. Не более 800 символов. Параметр появляется, если отмечена опция «Уведомлять о поступлении новых объявлений по SMS».

Шаблонные теги

Для работы с модулем «Объявления» служат следующие [шаблонные теги](#):

show_block – выводит последние объявления на всех страницах, кроме страницы объявлений, когда выводится список тех же объявлений, что и в функции.

Атрибуты:

- **count** – количество выводимых объявлений (по умолчанию 3);
- **site_id** – страницы, к которым прикреплен модуль. Идентификаторы страниц перечисляются через запятую. По умолчанию выбираются все страницы;
- **cat_id** – категории объявлений, если в настройках модуля отмечено «Использовать категории». Идентификаторы категорий перечисляются через запятую. По умолчанию категория не учитывается, выводятся все объявления;
- **sort** – сортировка объявлений: **date** – по дате (по умолчанию), **rand** – в случайном порядке;
- **only_ab** – выводить блок только на странице, к которой прикреплен модуль «Объявления»;
- **images** – количество изображений, прикрепленных к объявлениям;
- **images_variation** – тег размера изображений, задается в настройках модуля;
- **param** – значения дополнительных характеристик;

Пример:

Объявления обладают следующими характеристиками:

- тип: квартира, дом, участок – выпадающий список, номер 3;
- площадь – число, номер 10;
- новостройка – галочка, номер 16.

Значит значение атрибута *param3=5¶m3=6¶m10>50¶m16=0* расшифровывается как объявления о квартире и доме (5 и 6 номер), площадью более 50 кв.м, в новостройке.

— **template** – шаблон тега (файл *modules/ab/views/ab.view.show_block_template.php*; по умолчанию шаблон *modules/ab/views/ab.view.show_block.php*).

Пример:

```
<insert name="show_block" module="ab">
```

выведет 3 последние объявления сайта

```
<insert name="show_block" module="ab" count="5">
```

выведет 5 последних объявлений сайта

```
<insert name="show_block" module="ab" count="1" site_id="12">
```

выведет последнее объявление, прикрепленное к странице 12

show_block_rel – на странице объявления выводит похожие объявления. По умолчанию связи между объявлениями являются односторонними, это можно изменить, отметив опцию «В блоке похожих объявлений связь двусторонняя» в настройках модуля.

Атрибуты:

- **count** – количество выводимых объявлений (по умолчанию 3);
- **images** – количество изображений, прикрепленных к объявлениям;
- **images_variation** – тег размера изображений, задается в настройках модуля;
- **template** – шаблон тега (файл *modules/ab/views/ab.view.show_block_rel_template.php*; по умолчанию шаблон *modules/ab/views/ab.view.show_block_rel.php*).

Формат даты задается в настройках модуля.

Пример:

```
<insert name="show_block_rel" module="ab">
```

выведет 3 объявления, прикрепленные к текущей объявлению

show_form – выводит форму добавления сообщения. Для правильной работы тега должна существовать страница, к которой прикреплен модуль Объявления.

Атрибуты:

- **only_ab** – выводить форму поиска только на странице модуля;
- **site_id** – страницы, к которым прикреплен модуль. Идентификаторы страниц перечисляются через запятую. По умолчанию выбираются все страницы. Если задано несколько страниц, то в форме появляется выпадающий список «Раздел»;
- **cat_id** – категории объявлений, если в настройках модуля отмечено «Использовать категории». Идентификаторы категорий перечисляются через запятую. Можно указать значение **current**, тогда форма будет добавлять объявление в текущую (открытую) категорию или выводить поле «Категория», если ни одна категория не открыта. Если задано несколько категорий, то в форме появляется выпадающий список «Категория»;
- **template** – шаблон тега (файл *modules/ab/views/feedback.view.form_template.php*; по

умолчанию шаблон *modules/ab/views/ab.view.form.php*).

Пример:

```
<insert name="show_form" module="ab">
```

выведет форму добавления объявления

show_search – выводит форму поиска объявлений. Если для категорий прикреплены дополнительные характеристики, то поиск по ним производится только на странице категории.

Атрибуты:

— **only_ab** – выводить форму поиска только на странице модуля;

— **site_id** – страницы, к которым прикреплен модуль. Идентификаторы страниц перечисляются через запятую. По умолчанию выбираются все страницы. Если выбрано несколько страниц сайта, то в форме поиска появляется выпадающий список по выбранным страницам;

— **cat_id** – категории объявлений, если в настройках модуля отмечено «Использовать категории». Идентификаторы категорий перечисляются через запятую. Можно указать значение **current**, тогда поиск будет осуществляться по текущей (открытой) категории или по всем категориям, если ни одна категория не открыта. Если выбрано несколько категорий, то в форме поиска появится выпадающий список категорий, который будет подгружать прикрепленные к категориям характеристики. Можно указать значение **all**, тогда поиск будет осуществляться по всем категориям объявлений и в форме будут участвовать только общие характеристики;

— **ajax** – подгружать результаты поиска без перезагрузки страницы. Результаты подгружаются только если открыта страница со списком объявлений, иначе поиск работает обычным образом;

— **template** – шаблон тега (файл *modules/ab/views/ab.view.show_search_template.php*; по умолчанию шаблон *modules/ab/views/ab.view.show_search.php*).

Пример:

```
<insert name="show_search" module="ab">
```

выведет форму поиска по каталогу объявлений

Файлы

Модуль находится в папке *modules/ab* и устроен [классически](#).

1. *modules/ab/ab.php* – контроллер;
2. *modules/ab/ab.ajax.php* – обработка запроса на добавление объявления;

3. *modules/ab/ab.install.php* – установка модуля;
4. *modules/ab/ab.model.php* – модель;
5. *modules/ab/ab.rss.php* – RSS лента объявлений;
6. *modules/ab/ab.search.php* – настройки для поисковой индексации для модуля «Поиск»;
7. *modules/ab/ab.sitemap.php* – карта ссылок для модуля «Карта сайта»;
8. *modules/ab/admin/ab.admin.php* – редактирование объявлений;
9. *modules/ab/admin/ab.admin.category.php* – редактирование категорий объявлений;
10. *modules/ab/admin/ab.admin.config.php* – настройки модуля;
11. *modules/ab/admin/ab.admin.counter.php* – статистика просмотров;
12. *modules/ab/admin/ab.admin.js* – редактирование объявлений, JS-сценарий;
13. *modules/ab/admin/ab.admin.menu.php* – карта ссылок для модуля «Меню на сайте»;
14. *modules/ab/admin/ab.admin.param.php* – редактирование дополнительных характеристик объявлений;
15. *modules/ab/admin/ab.admin.param.js* – редактирование характеристик объявлений, JS-сценарий;
16. *modules/ab/img/** – изображения;
17. *modules/ab/js/ab.edit.js* – JS-сценарий формы редактирования объявления;
18. *modules/ab/js/ab.form.js* – JS-сценарий формы добавления объявления;
19. *modules/ab/js/ab.js* – JS-сценарий модуля;
20. *modules/ab/js/ab.show_search.js* – JS-сценарий формы поиска по объявлениям;
21. *modules/ab/views/ab.view.attachments.php* – шаблон прикрепленных к объявлению файлов в форме редактирования объявления;
22. *modules/ab/views/ab.view.edit.php* – шаблон формы редактирования объявления;
23. *modules/ab/views/ab.view.first_page.php* – шаблон вывода первой страницы модуля, если в настройках модуля подключен параметр «Использовать категории»;
24. *modules/ab/views/ab.view.form.php* – шаблон формы добавления объявления;
25. *modules/ab/views/ab.view.id.php* – шаблон страница объявления;
26. *modules/ab/views/ab.view.images.php* – шаблон прикрепленных к объявлению изображения, выводимых в форме редактирования объявления;
27. *modules/ab/views/ab.view.list.php* – шаблон список объявлений;
28. *modules/ab/views/ab.view.param.php* – шаблон дополнительных характеристик объявлений;

- 29. *modules/ab/views/ab.view.show_block.php* – шаблон блока объявлений;
- 30. *modules/ab/views/ab.view.show_block_rel.php* – шаблон блока похожих объявлений;
- 31. *modules/ab/views/ab.view.show_search.php* – шаблон форма поиска по объявлениям.

Теги

- [Подключение](#)
- [Как добавлять теги к модулю](#)
- [Административная часть](#)
- [Настройки](#)
- [Шаблонные теги](#)
- [Файлы](#)

Модуль относится к части CMS «Контент» и может быть установлен опционально.

Модуль «Теги» подключается к другим модулям. В стандартной комплектации diafan.CMS «Теги» подключены к следующим модулям: Фотогалерея, Новости, Статьи, Вопрос-Ответ, Магазин, Файловый архив. Включение тегов в рамках модуля настраивается в конфигурации конкретного модуля.

Подключение

Подключаемая часть – файл *modules/tags/tags.inc.php*. В нем описан класс *Tags_inc*. В модуле к объекту класса можно обратиться через переменную `$this->diafan->_tags`. Экземпляр класса создается при первом вызове переменной.

Методы

string **get**([integer \$element_id = 0], [string \$module_name = ''], [string \$site_id = 0]) – выводит подключенные к элементу теги.

- access: public
- integer \$element_id: номер элемента модуля
- string \$module_name: название модуля
- integer \$site_id: страница сайта, к которой прикреплен элемент

Пример:

```
// получаем теги статьи в файле modules/clauses/clauses.model.php
$tags = $this->diafan->_tags->get($id, 'clauses');
// выводим теги в шаблоне модуля в файле modules/clauses/clauses.view.php
echo $tags;
```

Как добавлять теги к модулю

В конфигурации модуля (*modules/модуль/admin/модуль.admin.config.php*) и в файле редактирования элементов модуля (*modules/модуль/admin/модуль.admin.php*) необходимо добавить параметр **'tags'**:

Пример:

```
public $variables = array(
    'main' => array(
        'tags' => 'module',
        ...
    ),
    ...
);
```

Страница тега в пользовательской части выводит список элементов других модулей, к которым тег подключен.

Данные о подключаемом элементе генерируются либо функцией *tags*, либо, если *tags* не определена, функцией *get_elements* из модели подключаемого модуля.

Пример:

В классе *Shop_model* файл *modules/shop/shop.model.php* определена функция *tags*. Она используется модулем «Теги» для формирования данных о товаре для списка тегов.

А вот для новостей функция *tags* не определена, поэтому данные о новости, к которой прикреплен тег, формируются с помощью функции *get_elements* класса *News_model* из файла *modules/news/news.model.php*.

Похожая ситуация с шаблонами вывода. Если в шаблоне модуля определен файл *modules/модуль/views/модуль.view.list_tags.php*, то он используется, иначе используется стандартный шаблон вывода списка элементов модуля – *modules/модуль/views/модуль.view.list.php*.

Пример:

В модуле «Фотогалерея» определен файл *modules/photo/views/photo.view.list_tags.php*, поэтому он используется в качестве шаблона вывода фотографий для списка элементов, прикрепленных к тегу.

В модуле «Статьи» не определен файл *modules/clauses/views/clauses.view.list_tags.php*, поэтому используется стандартный шаблон *modules/clauses/views/clauses.view.list.php*.

Административная часть

Теги [изменить](#)

 [Добавить тег](#)

	☐ Орел
	☐ Птица
	☐ Желтый
	☐ Мышь
	☐ Пингвин
	☐ Весна

В административной части модуля собираются все теги в одну таблицу, но при редактировании конкретного элемента модуля, к которому подключены теги, можно редактировать теги к нему. Теги можно сортировать по принадлежности к отдельному элементу.

Кнопка  служит для удаления тега. Порядок отображения тегов можно менять с помощью мыши простым перетаскиванием.

При удалении какого-либо элемента удаляются все теги, прикрепленные к нему.

Редактировать

Название

Псевдоссылка

Редирект на текущую страницу со
страницы:

Код ошибки

Changefreq

Priority

Сортировка: установить перед

[Орел](#)

Теги имеют следующие характеристики.

Основные:

- **Название** – Параметр имеет разные значения для разных языковых версий сайта.
- **Псевдоссылка** – ЧПУ (человеко-понятные url), адрес страницы вида:

<http://site.ru/pseudossylka/>. Смотрите параметры сайта.

- **Редирект на текущую страницу со страницы** – Позволяет делать редирект с указанной страницы на текущую.
- **Изображения** – Назначить тегу изображение.
- **Changefreq** – Вероятная частота изменения этой страницы. Это значение используется для генерирования файла *sitemap.xml*. Подробнее читайте в описании [XML-формата файла Sitemap](#) (SEO-специалисту).
- **Priority** – Приоритетность URL относительно других URL на Вашем сайте. Это значение используется для генерирования файла *sitemap.xml*. Подробнее читайте в описании [XML-формата файла Sitemap](#) (SEO-специалисту).
- **Сортировка: установить перед** – Редактирование порядка отображения пункта.
- **Индексирование для карты сайта** – Тег автоматически индексируется для карты сайта *sitemap.xml*.
- **Не показывать на карте сайта** – Скрывает отображение ссылки на новость в файле *sitemap.xml* и [модуле «Карта сайта»](#).

Настройки

- **Количество элементов на странице** – Количество одновременно выводимых новостей, товаров, фотографий и др., помеченных тегом в списке.
- **Использовать изображения** – Если отмечена, к тегам можно будет добавлять изображения.
- **Генерировать размеры изображений** – Размеры изображений, заданные в модуле «Изображения».
- **Отображение изображений в списке** – Параметр принимает значения:
 - нет (отключает отображение изображений в списке);
 - показывать одно изображение;
 - показывать все изображения. Параметр выводится, если отмечена опция «Использовать изображения».
- **Максимальный размер загружаемых файлов** – Параметр показывает максимально допустимый размер загружаемых файлов, установленный в настройках хостинга. Параметр выводится, если отмечена опция «Использовать изображения».
- **Применить настройки ко всем ранее загруженным изображениям** – Позволяет перекодировать размер уже загруженных изображений. Кнопка необходима, если изменены настройки размеров изображений. Параметр выводится, если отмечена опция «Использовать изображения».

Шаблонные теги

Для работы с модулем «Теги» служат следующие [шаблонные теги](#):

show_block – выводит список тегов.

Атрибуты:

— **template** – шаблон тега (файл *modules/tags/views/tags.view.show_block_template.php*; по умолчанию шаблон *modules/tags/views/tags.view.show_block.php*).

Пример:

```
<insert name="show_block" module="tags">
```

выведет облако тегов

Файлы

Модуль находится в папке *modules/tags*.

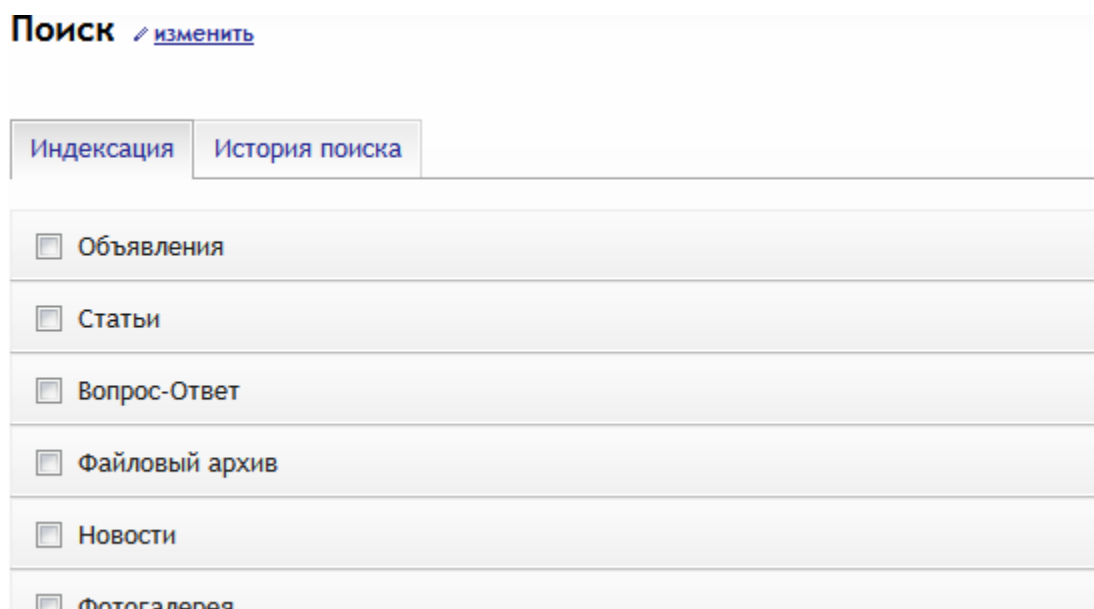
1. *modules/tags/admin/tags.admin.php* – редактирование тегов;
2. *modules/tags/admin/tags.admin.ajax.php* – обработка Ajax-запросов при работе с тегами в административной части;
3. *modules/tags/admin/tags.admin.config.php* – настройки модуля;
4. *modules/tags/admin/tags.admin.inc.php* – подключение модуля к административной части других модулей;
5. *modules/tags/admin/tags.admin.inc.js* – редактирование тегов, JS-сценарий;
6. *modules/tags/admin/tags.admin.js* – редактирование тегов, JS-сценарий;
7. *modules/tags/admin/tags.admin.view.php* – шаблон вывода тегов в административной части;
8. *modules/tags/tags.php* – контроллер;
9. *modules/tags/tags.inc.php* – подключение для работы с тегами;
10. *modules/tags/tags.install.php* – установка модуля;
11. *modules/tags/tags.model.php* – модель;
12. *modules/tags/tags.sitemap.php* – карта ссылок для модуля «Карта сайта»;
13. *modules/tags/views/tags.view.get.php* – шаблон списка прикрепленных к элементу тегов;
14. *modules/tags/views/tags.view.list.php* – шаблон списка элементов, к которым прикреплен тег;
15. *modules/tags/views/tags.view.show_block.php* – шаблон облака тегов.

Поиск

- [Индексация](#)
- [История поиска](#)
- [Настройки](#)
- [Шаблонные теги](#)
- [Файлы](#)

Модуль выводит форму для поиска по сайту и результаты поиска.

Индексация



Административная часть модуля нужна для ручной работы с поисковым индексом. Здесь можно переиндексировать контент всех или отдельных модулей, а также удалить весь индекс или индекс отдельных модулей. Ручная работа с индексом необходима только в следующих случаях:

- контент поступил извне: из предыдущей версии, где не было индексируемого поиска (4.x версия), импортирован из CSV-файла, добавлен вручную в базу данных;
- модуль «Поиск» был установлен на работающий сайт;
- из-за сбоя контент остался неиндексированным;
- необходимо, чтобы контент какого-либо модуля не участвовал в поисковой выдаче (в этом случае кроме удаления индекса для модуля надо еще удалить файл `modules/модуль/модуль.search.php`).

При добавлении/редактировании контента из административной части контент будет переиндексироваться автоматически без ручного вмешательства. Необходимость индексировать контент конкретного модуля и настройки индексации описаны в файле `modules/модуль/модуль.search.php`. Формат файла:

Пример:

```

class Модуль_search_config
{
    public $config = array(
        'таблица' => array(
            'fields' => array('название поля для индекса', ...),
            'name_field' => 'поле использующееся в качестве названия',
            'snippet_fields' => array('название поля, на основе которого формируется
снимет', ...),
            'snippet_length' => длина_снимета_в_символах,
            'rating' => рейтинг_модуля_в_поисковой_выдаче
        ),
        'таблица_2' => array(
            ...
        ),
        ...
    );
}

```

История поиска

История поиска

Индексация

История поиска

☐	12.04.2013 09:44	<u>заяц</u>
☐	12.04.2013 09:43	<u>мышь</u>

☐ Отметить все

Удалить ▾

Применить

На странице:

Изменить

Поисковые запросы имеют следующие характеристики:

Основные:

- **Запрос.**
- **Дата.**

Список последних поисковых запросов. Количество запросов в истории поиска определяется в настройках модуля.

Настройки

Настройки

Количество результатов на странице

10

Количество последних запросов в истории поиска

10

Искать все слова сразу

☐ ?

Сохранить

- **Количество результатов на странице** – Количество одновременно выводимых результатов поиска в списке.
- **Количество последних запросов в истории поиска.**
- **Искать все слова сразу** – Если не отмечено, ищет хотя бы одно слово.

Шаблонные теги

Для работы с модулем «Поиск по сайту» служат следующие [шаблонные теги](#):

show_search – выводит форму поиска по сайту.

Атрибуты:

— **button** – значение кнопки «Найти». Для неосновной языковой версии значение можно перевести в административной части в меню «Языки сайта» – «Перевод интерфейса»;

— **template** – шаблон тега (файл *modules/search/views/search.view.show_search_***template***.php*; по умолчанию шаблон *modules/search/views/search.view.show_search.php*).

Пример:

```
<insert name="show_search" module="search" button="найти">
```

выведет форму поиска по сайту. "button" - необязательный атрибут названия кнопки поиска

Файлы

1. *modules/search/admin/search.admin.php* – работа с поисковым индексом;
2. *modules/search/admin/search.admin.config.php* – настройки модуля;
3. *modules/search/admin/search.admin.history.php* – история поиска;
4. *modules/search/admin/search.admin.inc.php* – подключение модуля к административной части других модулей;
5. *modules/search/admin/search.admin.index.php* – создание удаление поискового индекса;
6. *modules/search/search.php* – контроллер;
7. *modules/search/search.install.php* – установка модуля;
8. *modules/search/search.model.php* – модель;
9. *modules/search/views/search.view.show.php* – шаблон результатов поиска по сайту;
10. *modules/search/views/search.view.show_search.php* – шаблон формы поиска по сайту;
11. *modules/search/views/search.view.show_search_top.php* – шаблон формы поиска по сайту, `template=top`.

Вопрос-Ответ

- [Вопросы - ответы](#)
- [Категории](#)
- [Настройки](#)
- [Шаблонные теги](#)
- [Файлы](#)

Модуль позволяет пользователям задавать вопросы на сайте, а администратору публиковать ответы на эти вопросы с возможностью отправки ответа на e-mail пользователя, задавшего вопрос.

Модуль относится к части CMS «Интерактив» и может быть установлен опционально.

Модуль «Вопрос-Ответ» можно прикреплять к различным страницам сайта. Если модуль прикреплен к нескольким страницам, в административной части модуля появляется специальный фильтр «Раздел сайта» с возможностью фильтрации вывода (списка вопросов, категорий и настроек) по принадлежности к странице.

Модуль состоит из двух частей: категории вопросов и вопросы-ответы.

Вопросы - ответы

Вопрос-Ответ [изменить](#)

Вопросы

Категории

Категория:

Все

+

[Добавить вопрос-ответ](#)

☐	11.04.2013	Требуются ли вам сотрудники...	(без ответа)
☐	10.04.2013	Выпускаете ли вы своих...	Мы их выпускаем - там свалку - там

В списке вопросы выводятся с приоритетом новизны.

Кнопка  служит для удаления вопроса. Кнопка  служит для быстрой публикации вопроса (или, наоборот, блокирования).

Редактировать 

Вопрос

Показывать на сайте ☒ ?

Часто задаваемый вопрос ☐ ?

Имя ?

Дата  ?

Email

Вопросы имеют следующие характеристики.

Основные:

- **Вопрос** – Вопрос, который задал пользователь в форме в пользовательской части сайта. Параметр имеет разные значения для разных языковых версий сайта.
- **Опубликовать на сайте** – Если отмечена, вопрос и ответ видны на сайте. Параметр имеет разные значения для разных языковых версий сайта.
- **Часто задаваемый вопрос** – Используется в шаблонной функции `show_block`.
- **Имя** – Имя отправителя вопроса. Параметр имеет разные значения для разных языковых версий сайта.
- **Дата и время** – Дата и время поступления вопроса в формате дд.мм.гггг чч:мм, приходит с сайта.
- **Email** – Электронный ящик получателя ответа.
- **Отправить ответ** – Если отмечена, после сохранения сообщения ответ будет послан на e-mail отправителя. Содержание письма, а также e-mail, указываемый в обратном адресе можно редактировать в настройках модуля. Письмо не может быть отправлено, если не заполнено текстовое поле для ответа, поле с вопросом или e-mail получателя.
- **Прикрепленные файлы.**
- **Похожие вопросы** – Выбор и добавление к текущему вопросу связей с другими вопросами. Похожие вопросы выводятся шаблонным тегом `show_block_rel`. По умолчанию связи между вопросами являются односторонними, это можно изменить, отметив опцию «В блоке похожих вопросов связь двусторонняя» в настройках модуля.
- **Теги** – Добавление тегов к вопросу. Можно добавить либо новый тег, либо открыть и выбрать из уже существующих тегов. Параметр выводится, если в настройках модуля включен параметр «Подключить теги».
- **Раздел сайта** – Перенос вопроса на другую страницу сайта, к которой прикреплен модуль. Параметр выводится, если в настройках модуля отключена опция «Использовать категории», если опция подключена, то раздел сайта задается такой же, как у основной категории.
- **Категория** – Категория, к которой относится вопрос. Возможно выбрать дополнительные категории, в которых вопрос также будет выводиться. Чтобы выбрать несколько категорий, удерживайте CTRL. Параметр выводится, если в настройках

модуля отмечена опция «Использовать категории».

- **Счетчик просмотров** – Количество просмотров на сайте текущего вопроса. Статистика ведется и параметр выводится, если в настройках модуля отмечена опция «Подключить счетчик просмотров».
- **Комментарии** – Комментарии, которые оставили пользователи к текущему вопросу. Параметр выводится, если в настройках модуля включен параметр «Показывать комментарии к вопросам».
- **Рейтинг** – Средний рейтинг, согласно голосованию пользователей сайта. Параметр выводится, если в настройках модуля включен параметр «Подключить рейтинг к вопросам».
- **Ответ** – Текстовое поле для ответа. Параметр имеет разные значения для разных языковых версий сайта.
- **Индексирование для поиска** – Вопрос автоматически индексируется для модуля «Поиск по сайту» при внесении изменений.
- **Индексирование для карты сайта** – Вопрос автоматически индексируется для карты сайта *sitemap.xml*.

Дополнительные:

- **Номер** – Номер элемента в БД (веб-мастеру и программисту).
- **Автор** – Пользователь, создавший вопрос в форме на сайте.
- **Заголовок окна в браузере, *тег Title*** – Если не заполнен, *тег Title* будет автоматически сформирован как «Название страницы – Название сайта» Параметр имеет разные значения для разных языковых версий сайта.
- **Ключевые слова, *тег Keywords*** – Если не заполнен, *тег Keywords* будет автоматически сформирован согласно шаблонам автоформирования из настроек модуля (SEO-специалисту). Параметр имеет разные значения для разных языковых версий сайта.
- **Описание, *тег Description*** – Если не заполнен, *тег Description* будет автоматически сформирован согласно шаблонам автоформирования из настроек модуля (SEO-специалисту). Параметр имеет разные значения для разных языковых версий сайта.
- **Псевдоссылка** – ЧПУ (человеко-понятные урл url), адрес страницы вида: *http://site.ru/pseudossylka/*. Смотрите параметры сайта.
- **Редирект на текущую страницу со страницы** – Позволяет делать редирект с указанной страницы на текущую.
- **Changefreq** – Вероятная частота изменения этой страницы. Это значение используется для генерирования файла *sitemap.xml*. Подробнее читайте в описании [XML-формата файла Sitemap](#) (SEO-специалисту).
- **Priority** – Приоритетность URL относительно других URL на Вашем сайте. Это значение используется для генерирования файла *sitemap.xml*. Подробнее читайте в описании [XML-формата файла Sitemap](#) (SEO-специалисту).
- **Период показа** – Если заполнить, текущий вопрос будет опубликована на сайте в указанный период. В иное время пользователи сайта вопрос не будут видеть, получая ошибку 404 «Страница не найдена» (администратору сайта).
- **Доступ** – Если отметить опцию «Доступ только», вопрос и ответ увидят только авторизованные на сайте пользователи, отмеченных типов. Не авторизованные, в том числе поисковые роботы, увидят «404 Страница не найдена» (администратору сайта).
- **Не показывать на карте сайта** – Скрывает отображение ссылки на вопрос в файле *sitemap.xml* и [модуле «Карта сайта»](#).
- **Шаблон страницы** – Возможность подключить для страницы вопроса шаблон сайта отличный от основного (*themes/site.php*). Все шаблоны для сайта должны храниться в

папке *themes* с расширением *.php* (например, *themes/dizain_so_slajdom.php*).

Подробнее в [разделе «Шаблоны сайта»](#). (веб-мастеру и программисту, не меняйте этот параметр, если не уверены в результате!).

- **Шаблон модуля** – Шаблон вывода контента модуля на странице отдельного вопроса (веб-мастеру и программисту, не меняйте этот параметр, если не уверены в результате!).
- **Редактор** – Изменяется после первого сохранения. Показывает, кто из администраторов сайта первый правил текущую страницу.
- **Время последнего изменения** – Изменяется после сохранения элемента. Отдается в заголовке *Last Modify*.

Категории

Вопрос-Ответ [изменить](#)

Вопросы Категории

[Добавить категорию](#)

Общие вопросы

Вопросы про диких макак

☐ Отметить все Применить На странице: 100

Категории вопросов могут иметь неограниченное число подкатегорий. При необходимости категории можно не использовать. Для этого нужно отключить параметр «Использовать категории» в настройках модуля.

Если перед названием категории имеется знак +, то этой категории принадлежат подкатегории. Добавить подкатеорию к категории можно нажав на знак  рядом с названием категории.

Порядок отображения категорий можно менять с помощью мыши, просто перетаскивая их.

Кнопка  служит для удаления категории. Кнопка  служит для быстрой публикации категории (или, наоборот, блокирования). При удалении категории удаляются все вложенные подкатегории и вопросы.

Редактировать 

Название

Показывать на сайте ☒

Создать пункт в меню ☐ Меню верхнее
☐ Меню каталог товаров
☐ Меню слева

Анонс

HTML |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |

автоматически сформирован как «Название категории – Название страницы – Название сайта», либо согласно шаблонам автоформирования из настроек модуля (SEO-специалисту). Параметр имеет разные значения для разных языковых версий сайта.

- **Ключевые слова, тег Keywords** – Если не заполнен, тег *Keywords* будет автоматически сформирован согласно шаблонам автоформирования из настроек модуля (SEO-специалисту). Параметр имеет разные значения для разных языковых версий сайта.
- **Описание, тег Description** – Если не заполнен, тег *Description* будет автоматически сформирован согласно шаблонам автоформирования из настроек модуля (SEO-специалисту). Параметр имеет разные значения для разных языковых версий сайта.
- **Псевдоссылка** – ЧПУ, т.е. адрес страницы вида: <http://site.ru/pseudossylka/>. Смотрите параметры сайта (SEO-специалисту).
- **Редирект на текущую страницу со страницы** – Позволяет делать редирект с указанной страницы на текущую.
- **Changefreq** – Вероятная частота изменения этой страницы. Это значение используется для генерирования файла *sitemap.xml*. Подробнее читайте в описании [XML-формата файла Sitemap](#) (SEO-специалисту).
- **Priority** – Приоритетность URL относительно других URL на Вашем сайте. Это значение используется для генерирования файла *sitemap.xml*. Подробнее читайте в описании [XML-формата файла Sitemap](#) (SEO-специалисту).
- **Раздел сайта** – Перенос категории на другую страницу сайта, к которой прикреплен модуль (администратору сайта).
- **Сортировка: установить перед** – Редактирование порядка следования категории в списке. Поле доступно для редактирования только для категорий, отображаемых на сайте.
- **Вложенность: принадлежит** – Перемещение текущей категории и всех её подкатегорий в принадлежность другой категории (администратору сайта).
- **Доступ** – Если отметить опцию «Доступ только», категорию увидят только авторизованные на сайте пользователи, отмеченных типов. Не авторизованные, в том числе поисковые роботы, увидят «404 Страница не найдена» (администратору сайта).
- **Не показывать на карте сайта** – Скрывает отображение ссылки на категорию в файле *sitemap.xml* и [модуле «Карта сайта»](#).
- **Шаблон страницы** – Возможность подключить для страницы категории шаблон сайта отличный от основного (*themes/site.php*). Все шаблоны для сайта должны храниться в папке *themes* с расширением *.php* (например, *themes/dizain_so_slajdom.php*). Подробнее в [разделе «Шаблоны сайта»](#). (веб-мастеру и программисту, не меняйте этот параметр, если не уверены в результате!).
- **Шаблон модуля** – Шаблон вывода контента модуля на странице списка вопросов в категории (веб-мастеру и программисту, не меняйте этот параметр, если не уверены в результате!).
- **Шаблон модуля для вложенных вопросов** – Шаблон вывода контента модуля на странице отдельного вопроса, вложенного в текущую категорию (веб-мастеру и программисту, не меняйте этот параметр, если не уверены в результате!).
- **Редактор** – Изменяется после первого сохранения. Показывает, кто из администраторов сайта первый правил текущую страницу.
- **Время последнего изменения** – Изменяется после сохранения элемента. Отдается в заголовке *Last Modify*.

Настройки

Настройки

Количество вопросов на странице	10
Только для зарегистрированных пользователей	☐
Использовать категории	☒
Количество вопросов в списке категорий	3
Количество вопросов в списке вложенной категории	
Показывать вопросы подкатегорий	☐

- **Количество вопросов на странице** – Количество одновременно выводимых вопросов в списке.
- **Только для зарегистрированных пользователей** – Вопросы могут добавлять только зарегистрированные пользователи.
- **Использовать категории** – Позволяет включить/отключить категории вопросов.
- **Количество вопросов в списке категорий** – Количество вопросов, выводимых в списке категорий на главной странице модуля. Параметр появляется, если отмечена опция «Использовать категории».
- **Количество вопросов в списке вложенной категории** – Для первой страницы модуля и для страницы категории. Параметр появляется, если отмечена опция «Использовать категории».
- **Показывать вопросы подкатегорий** – Если отмечена, в списке вопросов категории будут отображаться вопросы из всех вложенных категорий. Параметр появляется, если отмечена опция «Использовать категории».
- **Счетчик просмотров** – Позволяет считать количество просмотров отдельного вопроса.
- **Выводить счетчик на сайте** – Позволяет вывести на сайте количество просмотров отдельного вопроса. Параметр выводится, если отмечена опция «Счетчик просмотров». Параметр появляется, если отмечена опция «Счетчик просмотров».
- **Использовать защитный код (капчу)** – Для добавления вопроса пользователь должен ввести защитный код.
- **Формат даты** – Позволяет настроить отображение даты в модуле. Возможные значения:
 - 01.05.2013,
 - 01.05.2013 14:45,
 - 1 мая 2013 г.,
 - 1 мая,
 - 1 мая 2013, понедельник,
 - вчера 15:30,
 - не отображать.
- **Сообщение после отправки** – Сообщение, получаемое пользователем при удачной загрузке вопроса, допускаются HTML-теги для оформления сообщения. Параметр

имеет разные значения для разных языковых версий сайта.

Пример:

```
<div align="center"><b>Спасибо за ваше сообщение!</b><br>Наш консультант  
подберет необходимую информацию, после чего ваш вопрос и ответ на него будут  
опубликованы на этой странице.</div>
```

- **Ваше сообщение уже имеется в базе** – Сообщение, получаемое пользователем при повторной попытке отправить вопрос. Параметр имеет разные значения для разных языковых версий сайта.
- **Разрешить добавление файлов** – Позволяет пользователям прикреплять файлы к вопросам.
- **Максимальное количество добавляемых файлов** – Количество добавляемых файлов. Если значение равно нулю, то форма добавления файлов не выводится.
- **Доступные типы файлов (через запятую).**

Пример:

doc, gif, jpg, mpg, pdf, png, txt, zip

- **Распознавать изображения** – Позволяет прикрепленные к вопросу файлы в формате JPEG, GIF, PNG отображать как изображения.
- **Размер для большого изображения** – Размер изображения, отображаемый в пользовательской части сайта при увеличении изображения предпросмотра.
- **Размер для маленького изображения** – Размер изображения предпросмотра.
- **Использовать анимацию при увеличении изображений** – Параметр добавляет JavaScript код, позволяющий включить анимацию при увеличении изображений. Параметр выводится, если отмечена опция «Распознавать изображения».
- **Максимальный размер загружаемых файлов** – Параметр показывает максимально допустимый размер загружаемых файлов, установленный в настройках хостинга.
- **Тема письма для ответа** – Ответ отправляется пользователю, если при редактировании вопроса заполнены поля «E-mail», «Вопрос», «Ответ» и «Отправить ответ». Можно добавлять:
 - %title – название сайта,
 - %url – адрес сайта (например, site.ru).

Параметр имеет разные значения для разных языковых версий сайта.

Пример:

%title (%url). Вопрос-Ответ.

- **Сообщение для ответа** – Можно добавлять:
 - %name – имя пользователя,
 - %title – название сайта,
 - %url – адрес сайта (например, site.ru),
 - %question – вопрос,
 - %answer – ответ.

Параметр имеет разные значения для разных языковых версий сайта.

Пример:

Здравствуйте, %name!
Вы задали вопрос на сайте %title

(%url).
Вопрос: %question
Ответ: %answer.

- **Е-mail, указываемый в обратном адресе пользователю** – Возможные значения:
 - е-mail, указанный в параметрах сайта;
 - другой (при выборе этого значения появляется дополнительное поле **впишите е-mail**).
- **Уведомлять о поступлении новых вопросов на е-mail** – Возможность уведомления администратора о поступлении новых вопросов из формы в пользовательской части сайта.
- **Е-mail для уведомлений администратора** – Возможные значения:
 - е-mail, указанный в параметрах сайта;
 - другой (при выборе этого значения появляется дополнительное поле **впишите е-mail**).

Параметр появляется, если отмечена опция «Уведомлять о поступлении новых вопросов на е-mail».

- **Тема письма для уведомлений** – Можно добавлять:
 - %title – название сайта,
 - %url – адрес сайта (например, site.ru).

Параметр появляется, если отмечена опция «Уведомлять о поступлении новых вопросов на е-mail».

Пример:

%title (%url). Новый вопрос в рубрике Вопрос-Ответ.

- **Сообщение для уведомлений** – Можно добавлять:
 - %title – название сайта,
 - %url – адрес сайта (например, site.ru),
 - %question – вопрос,
 - %name – имя пользователя,
 - %email – е-mail пользователя,
 - %files – название прикрепляемых файлов.

Параметр появляется, если отмечена опция «Уведомлять о поступлении новых вопросов на е-mail».

Пример:

Здравствуйте, администратор сайта %title (%url)!
В рубрике Вопрос-Ответ появился новый вопрос:
%question.
%name
%email. Если включен параметр «Разрешить добавление файлов», то добавляется строка:

Прикрепленные файлы: %files.

- **Уведомлять о поступлении новых вопросов по SMS** – Возможность отправлять SMS администратору при создании вопроса на сайте. Параметр можно подключить, если в [Параметрах сайта](#) настроены SMS-уведомления.
- **Номер телефона в федеральном формате** – Номер телефона для SMS-уведомлений администратора о новом вопросе. Параметр появляется, если отмечена опция «Уведомлять о поступлении новых вопросов по SMS».
- **Сообщение для уведомлений** – Текст сообщения для SMS-уведомлений администратора о новом вопросе. Не более 800 символов. Параметр появляется, если отмечена опция «Уведомлять о поступлении новых вопросов по SMS».
- **Показывать комментарии к вопросам** – Подключение модуля «Комментарии».

Параметр не будет включен, если модуль «Комментарии» не установлен.

Подробности см. в разделе [модуль Комментарии](#).

- **Показывать комментарии к категориям** – Подключение модуля «Комментарии» к категориям вопросов. Параметр не будет включен, если модуль «Комментарии» не установлен. Подробности см. в разделе [модуль Комментарии](#).
- **Подключить теги** – Подключение модуля «Теги». Параметр не будет включен, если модуль «Теги» не установлен. Подробности см. в разделе [модуль «Теги»](#).
- **Показывать рейтинг вопросов** – Подключение модуля «Рейтинг». Параметр не будет включен, если модуль «Рейтинг» не установлен. Подробности см. в разделе [модуль «Рейтинг»](#).
- **Подключить рейтинг к категориям** – Подключение модуля «Рейтинг» к категориям. Параметр не будет включен, если модуль «Рейтинг» не установлен. Подробности см. в разделе [модуль «Рейтинг»](#).
- **Подключить перелинковку** – Отображение перелинковки в модуле. Подробности см. в разделе [модуль «Перелинковка»](#).
- **В блоке похожих вопросов связь двусторонняя** – Позволяет установленную в вопросе связь с другим вопросом использовать в двух направлениях.
- **Шаблон для автоматического генерирования Title** – Если шаблон задан, то заголовок автоматически генерируется по шаблону. В шаблон можно добавить:
 - %title – заданный заголовок,
 - %name – название,
 - %category – название категории,
 - %parent_category – название категории верхнего уровня (SEO-специалисту).Параметр имеет разные значения для разных языковых версий сайта.
- **Шаблон для автоматического генерирования Title для категории** – Если шаблон задан, то заголовок автоматически генерируется по шаблону. В шаблон можно добавить:
 - %title – заданный заголовок,
 - %name – название категории,
 - %parent – название категории верхнего уровня,
 - %page – страница (текст можно поменять в интерфейсе «Языки сайта» – «Перевод интерфейса») (SEO-специалисту).

Параметр имеет разные значения для разных языковых версий сайта. Параметр появляется, если отмечена опция «Использовать категории».

- **Шаблон для автоматического генерирования Keywords** – Если шаблон задан, то поле *Keywords* автоматически генерируется по шаблону. В шаблон можно добавить:
 - %keywords – заданные ключевые слова,
 - %name – название,
 - %category – название категории,
 - %parent_category – название категории верхнего уровня (SEO-специалисту).Параметр имеет разные значения для разных языковых версий сайта.
- **Шаблон для автоматического генерирования Keywords для категории** – Если шаблон задан, то поле *Keywords* автоматически генерируется по шаблону. В шаблон можно добавить:
 - %keywords – заданные ключевые слова,
 - %name – название категории,
 - %parent – название категории верхнего уровня (SEO-специалисту).

Параметр имеет разные значения для разных языковых версий сайта. Параметр

появляется, если отмечена опция «Использовать категории».

- **Шаблон для автоматического генерирования Description** – Если шаблон задан, то поле *Description* автоматически генерируется по шаблону. В шаблон можно добавить:
 - %descr – заданное описание,
 - %name – название,
 - %category – название категории,
 - %parent_category – название категории верхнего уровня (SEO-специалисту).

Параметр имеет разные значения для разных языковых версий сайта.

- **Шаблон для автоматического генерирования Description для категории** – Если шаблон задан, то поле *Description* автоматически генерируется по шаблону. В шаблон можно добавить:
 - %descr – заданное описание,
 - %name – название категории,
 - %parent – название категории верхнего уровня.

Параметр имеет разные значения для разных языковых версий сайта. Параметр появляется, если отмечена опция «Использовать категории».

- **Шаблон для списка элементов** – По умолчанию *modules/faq/views/faq.view.list.php*. Параметр для разработчиков! Не устанавливайте, если не уверены в результате.
- **Шаблон для первой страницы модуля (если подключены категории)** – По умолчанию *modules/faq/views/faq.view.fitst_page.php*. Параметр для разработчиков! Не устанавливайте, если не уверены в результате.
- **Шаблон для страницы элемента** – По умолчанию, *modules/faq/views/faq.view.id.php*. Параметр для разработчиков! Не устанавливайте, если не уверены в результате.
- **Отдельный пункт в меню администрирования для каждого раздела сайта** – Если модуль подключен к нескольким страницам сайта, отметка данного параметра выведет несколько пунктов в меню административной части для удобства быстрого доступа (администратору сайта).
- **Индексирование для карты сайта** – При изменении настроек, влияющих на отображение страницы, модуль автоматически переиндексируется для карты сайта *sitemap.xml*.

Шаблонные теги

Для работы с модулем «Вопрос-ответ» служат следующие [шаблонные теги](#):

show_block – выводит последние вопросы. На странице вопросов, когда выводится список аналогичных вопросов, блок не выводится.

Атрибуты:

— **count** – количество выводимых вопросов (по умолчанию 3);

— **site_id** – страницы, к которым прикреплен модуль. Идентификаторы страниц перечисляются через запятую. По умолчанию выбираются все страницы;

— **cat_id** – категории вопросов, если в настройках модуля отмечено «Использовать категории». Идентификаторы категорий перечисляются через запятую. По умолчанию категория не учитывается, выводятся все вопросы;

— **often** – часто задаваемые вопросы (если often=1, выводятся вопросы с пометкой

«Часто задаваемый вопрос»);

— **sort** – сортировка вопросов: **date** – по дате (по умолчанию), **rand** – в случайном порядке;

— **template** – шаблон тега (файл *modules/faq/views/faq.view.show_block_template.php*; по умолчанию шаблон *modules/faq/views/faq.view.show_block.php*).

Формат даты задается в настройках модуля.

Пример:

```
<insert name="show_block" module="faq">
```

выведет 3 последних активных вопроса сайта с ответами

```
<insert name="show_block" module="faq" count="5">
```

выведет в ячейке таблицы 5 последних активных вопросов сайта с ответами

```
<insert name="show_block" module="faq" often="1">
```

выведет часто задаваемые вопросы

show_block_rel – на странице вопроса выводит похожие вопросы. По умолчанию связи между вопросами являются односторонними, это можно изменить, отметив опцию «В блоке похожих вопросов связь двусторонняя» в настройках модуля.

Атрибуты:

— **count** – количество выводимых вопросов (по умолчанию 3);

— **template** – шаблон тега (файл *modules/faq/views/faq.view.show_block_rel_template.php*; по умолчанию шаблон *modules/faq/views/faq.view.show_block_rel.php*).

Формат даты задается в настройках модуля.

Пример:

```
<insert name="show_block_rel" module="faq">
```

выведет 3 вопроса, прикрепленные к текущему вопросу

show_form – выводит форму добавления вопроса. Для правильной работы тега должна существовать страница, к которой прикреплен модуль Вопрос-Ответ.

Атрибуты:

- **site_id** – страница, к которой прикреплен модуль, по умолчанию выбирается одна страница;
- **cat_id** – категория вопросов (id категории, по умолчанию выбирается одна категория), если в настройках модуля отмечено «Использовать категории»;
- **template** – шаблон тега (файл *modules/faq/views/faq.view.form_***template***.php*; по умолчанию шаблон *modules/faq/views/faq.view.form.php*).

Пример:

```
<insert name="show_form" module="faq">
```

выведет форму добавления вопроса

Файлы

Модуль находится в папке *modules/faq* и устроен [классически](#).

1. *modules/faq/admin/faq.admin.php* – редактирование вопросов;
2. *modules/faq/admin/faq.admin.category.php* – редактирование категорий вопросов;
3. *modules/faq/admin/faq.admin.config.php* – настройки модуля;
4. *modules/faq/admin/faq.admin.count.php* – количество неотвеченных сообщений из формы вопрос-ответ для меню административной панели;
5. *modules/faq/admin/faq.admin.counter.php* – статистика просмотров;
6. *modules/faq/admin/faq.admin.menu.php* – карта ссылок для модуля «Меню на сайте»;
7. *modules/faq/faq.php* – контроллер;
8. *modules/faq/faq.ajax.php* – обработка запроса при отправке сообщения из формы;
9. *modules/faq/faq.install.php* – установка модуля;
10. *modules/faq/faq.model.php* – модель;
11. *modules/faq/faq.search.php* – настройки для поисковой индексации для модуля «Поиск»;
12. *modules/faq/faq.sitemap.php* – карта ссылок для модуля «Карта сайта»;
13. *modules/faq/views/faq.view.first_page.php* – шаблон первой страницы модуля, если в настройках модуля подключен параметр «Использовать категории»;
14. *modules/faq/views/faq.view.form.php* – шаблон формы добавления вопроса;
15. *modules/faq/views/faq.view.id.php* – шаблон страницы вопроса;
16. *modules/faq/views/faq.view.list.php* – шаблон списка вопросов и ответов;

17. *modules/faq/views/faq.view.show_block.php* – шаблон блока вопросов и ответов;

18. *modules/faq/views/faq.view.show_block_rel.php* – шаблон блока похожих вопросов и ответов.

Обратная связь

- [Сообщения](#)
- [Конструктор формы](#)
- [Настройки](#)
- [Шаблонные теги](#)
- [Файлы](#)

Модуль позволяет пользователям оставлять сообщения на сайте и отвечать отправителям в административной части модуля. Модуль относится к части CMS «Интерактив» и может быть установлен опционально.

Модуль «Обратная связь» можно присоединить к различным страницам сайта. Если модуль прикреплен к нескольким страницам, в административной части модуля выводится фильтр «Раздел сайта», позволяющий отобразить сообщения и настройку модуля для одной страницы.

Сообщения

Обратная связь [изменить](#)

Сообщения

Конструктор формы

☐	11.04.2013 11:56	Редактировать	Вioletta, Надо мной посмея больше, пока она не извини
☐	10.04.2013 11:56	Редактировать	Георг, Были у вас на прошл кролик нам на ногу наступил
☐	09.04.2013 11:56	Редактировать	Георгий, Спасибо, что вы ес

☐ [Отметить все](#) [Удалить](#) [Применить](#) На странице: [Изменить](#)

В списке сообщения сортируются в порядке добавления пользователями.

Кнопка  служит для удаления сообщения.

Редактировать

Дата ?

Раздел сайта

Автор

Ваше имя

Ваш e-mail

☐ Отправить ответ

Надо мной посмеялась ваша ехидна, не больше, пока она не извинится!

Модуль имеет следующие характеристики.

Основные:

- **Дата** – Дата поступления сообщения в формате дд.мм.гггг чч:мм.
- **Раздел сайта** – Принадлежность к странице сайта.
- **Язык интерфейса** – Языковая версия сайта, на которой находился пользователь, когда писал сообщение.
- **Автор** – Пользователь, создавший сообщение в форме на сайте.
- **Конструктор формы** – Поля, добавленные в конструкторе формы.
- **Отправить ответ** – Возможность отправить ответ на e-mail, указанный пользователем. Содержание письма, а также e-mail, указываемый в обратном адресе можно редактировать в конфигурации модуля. Письмо не может быть отправлено, если не заполнено текстовое поле для ответа, поле с вопросом или e-mail получателя. Поле «Отправить ответ» появляется, если в конструкторе формы существует поле с типом «электронный ящик».
- **Отвечающий** – Пользователь, первый отредактировавший или создавший сообщение в административной части.
- **Ответ** – Текстовое поле для ответа.

При поступлении вопроса из формы в пользовательской части сайта может быть отправлено уведомительное письмо администратору. Тема письма, содержание и e-mail настраиваются в конфигурации модуля.

Если есть дополнительно поле с типом электронный ящик, то в административной части можно ответить на сообщение, отправив письмо на указанный ящик. Тема письма, содержание и обратный адрес настраиваются в конфигурации модуля.

Конструктор формы

На разных страницах сайта структура формы различная. Это обеспечивает конструктор формы.

Порядок отображения полей можно редактировать с помощью мыши и кнопки . Кнопка  служит для удаления полей.

Конструктор – это дополнительные поля формы обратной связи. Можно добавлять поля общие (для всех разделов) или присваивать поле отдельной форме на отдельной странице.

Дополнительные поля имеют следующие параметры.

Основные:

- **Название** – Параметр имеет разные значения для разных языковых версий сайта.
- **Раздел сайта** – Принадлежность к странице сайта.

- **Сортировка: установить перед** – Редактирование порядка следования характеристики в списке
- **Обязательно для заполнения.**
- **Тип** – Возможные значения:
 - строка,
 - число,
 - дата,
 - дата и время,
 - текстовое поле,
 - галочка,
 - выпадающий список,
 - список с выбором нескольких значений,
 - электронный ящик,
 - заголовок группы характеристик,
 - файлы,
 - изображения.
- **Максимальное количество добавляемых файлов** – Количество добавляемых файлов. Если значение равно нулю, то форма добавления файлов не выводится. Параметр выводится, если тип характеристики задан как «файлы».
- **Доступные типы файлов (через запятую)** – Параметр выводится, если тип характеристики задан как «файлы».

Пример:

doc, gif, jpg, mpg, pdf, png, txt, zip

- **Распознавать изображения** – Позволяет прикрепленные файлы в формате JPEG, GIF, PNG отображать как изображения. Параметр выводится, если тип характеристики задан как «файлы».
- **Размер для большого изображения** – Размер изображения, отображаемый в пользовательской части сайта при увеличении изображения предпросмотра. Параметр выводится, если тип характеристики задан как «файлы» и отмечена опция «Распознавать изображения».
- **Размер для маленького изображения** – Размер изображения предпросмотра. Параметр выводится, если тип характеристики задан как «файлы» и отмечена опция «Распознавать изображения».
- **Использовать анимацию при увеличении изображений** – Параметр добавляет JavaScript код, позволяющий включить анимацию при увеличении изображений. Параметр выводится, если отмечена опция «Распознавать изображения». Параметр выводится, если тип характеристики задан как «файлы» и отмечена опция «Распознавать изображения».
- **Максимальный размер загружаемых файлов** – Параметр показывает максимально допустимый размер загружаемых файлов, установленный в настройках хостинга. Параметр выводится, если тип характеристики задан как «файлы».
- **Генерировать размеры изображений** – Размеры изображений, заданные в модуле «Изображения». Параметр выводится, если тип характеристики задан как «изображение».
- **Значения** – Появляется для полей с типом «галочка», «выпадающий список» и «список с выбором нескольких значений»
- **Описание** – Параметр имеет разные значения для разных языковых версий сайта.

Если выбран тип «выпадающий список» или «список с выбором нескольких значений», то появятся дополнительные поля со значениями. Добавить значение можно нажав 🗲️ внизу последнего поля, сортировать значения можно с помощью мыши и кнопки 🗑️. Кнопка ✖️ служит для удаления значений.

Настройки

Настройки

Использовать защитный код (капчу) ☐

Сообщение после отправки

`<div align="center">Спасибо за вас
</div>`

Тема письма для ответа

`Здравствуйте!
Вы оставили сообщени
обратной связи на сайте %title (%url)`

- **Использовать защитный код (капчу)** – Для добавления сообщения пользователь должен ввести защитный код.
- **Сообщение после отправки** – Сообщение, получаемое пользователем при удачной загрузки вопроса, допускаются HTML-теги для оформления сообщения. Параметр имеет разные значения для разных языковых версий сайта.

Пример:

```
<div align="center"><b>Спасибо за ваше сообщение!</b></div>
```

- **Тема письма для ответа** – Можно добавлять:
 - %title – название сайта,
 - %url – адрес сайта (например, site.ru).

Параметр имеет разные значения для разных языковых версий сайта.

Пример:

```
%title (%url). Обратная связь
```

- **Сообщение для ответа** – Можно добавлять:
 - %title – название сайта,
 - %url – адрес сайта (например, site.ru),
 - %message – сообщение,
 - %answer – ответ.

Параметр имеет разные значения для разных языковых версий сайта.

Пример:

Здравствуйте!
Вы оставили сообщение в форме обратной связи на сайте %title (%url).
Сообщение: %message
Ответ: %answer.

- **Е-mail, указываемый в обратном адресе пользователю** – Возможные значения:
 - е-mail, указанный в параметрах сайта;
 - другой (при выборе этого значения появляется дополнительное поле **впишите е-mail**).
- **Уведомлять о поступлении новых сообщений на е-mail** – Возможность уведомления администратора о поступлении новых сообщений из формы в пользовательской части сайта.
- **Е-mail для уведомлений администратора** – Возможные значения:
 - е-mail, указанный в параметрах сайта;
 - другой (при выборе этого значения появляется дополнительное поле **впишите е-mail**).

Параметр появляется, если отмечена опция «Уведомлять о поступлении новых сообщений на е-mail».

- **Тема письма для уведомлений** – Можно добавлять:
 - %title – название сайта,
 - %url – адрес сайта (например, site.ru).

Параметр появляется, если отмечена опция «Уведомлять о поступлении новых сообщений на е-mail».

Пример:

%title (%url). Новое сообщение в рубрике Обратная связь

- **Сообщение для уведомлений** – Можно добавлять:
 - %title – название сайта,
 - %url – адрес сайта (например, site.ru)
 - %message – вопрос.

Параметр появляется, если отмечена опция «Уведомлять о поступлении новых сообщений на е-mail».

Пример:

Здравствуйте, администратор сайта %title (%url)!
В рубрике Обратная связь появилось новое сообщение: %message.

- **Уведомлять о поступлении новых сообщений по SMS** – Возможность отправлять SMS администратору при поступлении сообщения. Параметр можно подключить, если в [Параметрах сайта](#) настроены SMS-уведомления.
- **Номер телефона в федеральном формате** – Номер телефона для SMS-уведомлений администратора о новом сообщении. Параметр появляется, если отмечена опция «Уведомлять о поступлении новых сообщений по SMS».
- **Сообщение для уведомлений** – Текст сообщения для SMS-уведомлений администратора о новом сообщении. Не более 800 символов. Параметр появляется, если отмечена опция «Уведомлять о поступлении новых сообщений по SMS».
- **Отдельный пункт в меню администрирования для каждого раздела сайта** – Если модуль подключен к нескольким страницам сайта, отметка данного параметра выведет несколько пунктов в меню административной части для удобства быстрого доступа (администратору сайта).

Шаблонные теги

Для работы с модулем «Обратная связь» служат следующие [шаблонные теги](#):

show_form – выводит форму добавления сообщения. Для правильной работы тега должна существовать страница, к которой прикреплен модуль Обратная связь.

Атрибуты:

— **site_id** – страница, к которой прикреплен модуль, по умолчанию выбирается одна страница;

— **template** – шаблон тега (файл *modules/feedback/views/feedback.view.form_template.php*; по умолчанию шаблон *modules/feedback/views/feedback.view.form.php*).

Пример:

```
<insert name="show_form" module="feedback">
```

выведет форму добавления сообщения

Файлы

1. *modules/feedback/admin/feedback.admin.php* – редактирование сообщений из формы обратной связи;
2. *modules/feedback/admin/feedback.admin.config.php* – настройки модуля;
3. *modules/feedback/admin/feedback.admin.count.php* – количество неотвеченных сообщений из формы обратной связи для меню административной панели;
4. *modules/feedback/admin/feedback.admin.param.php* – конструктор формы обратной связи;
5. *modules/feedback/feedback.php* – контроллер;
6. *modules/feedback/feedback.ajax.php* – обработка Ajax-запроса;
7. *modules/feedback/feedback.install.php* – установка модуля;
8. *modules/feedback/feedback.model.php* – модель;
9. *modules/feedback/views/feedback.view.form.php* – шаблон формы добавления сообщения в обратной связи;
10. *modules/feedback/views/feedback.view.images.php* – шаблон вывода прикрепленных к сообщению изображений.

Форум

- [Категории](#)
- [Сообщения](#)
- [Настройки](#)
- [Шаблонные теги](#)
- [Файлы](#)

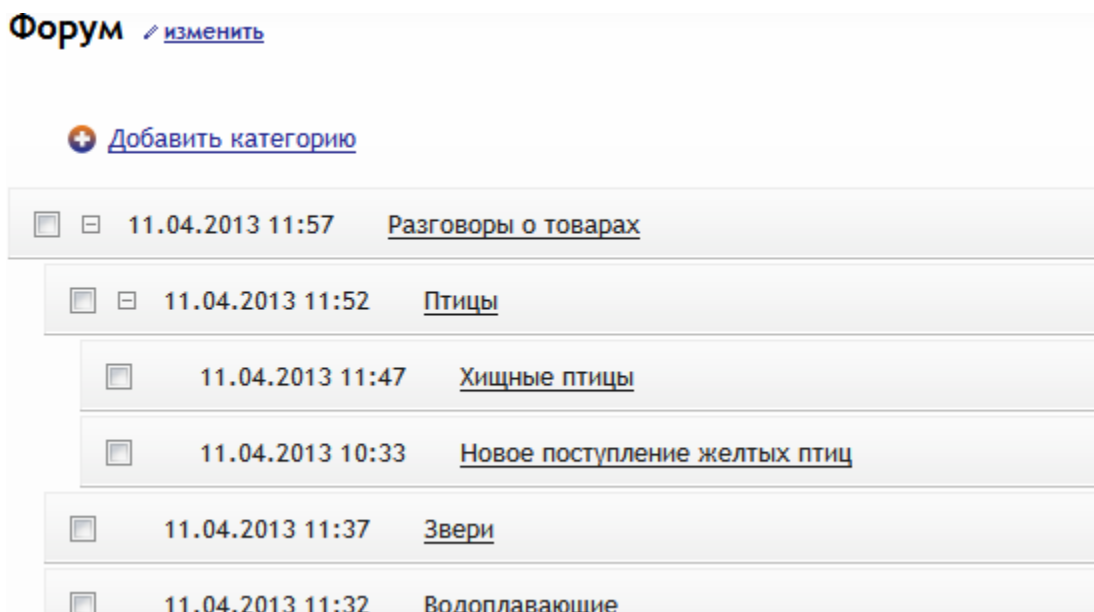
Модуль относится к части CMS «Интерактив» и может быть установлен опционально.

Для модуля «Форум» предусмотрен особый тип права доступа «Модераторы». Действия модератора можно подключить типу пользователя (*Пользователи сайта – Права доступа*) во вкладке «Права для пользовательской части».

Основная часть для работы с форумом – пользовательская. Пользователи добавляют темы и сообщения (в соответствии с конфигурацией модуля). Модераторы могут удалять, блокировать и редактировать темы и сообщения.

Административная часть модуля состоит из двух частей: категории и сообщения.

Категории



Категории формируют структуру форума:

- первый уровень – логические блоки,
- второй уровень – разделы;
- третий уровень – темы.

Прежде чем запустить форум, администратор должен создать структуру форума.

Логические блоки и разделы редактируются обычным образом – нажатие на название категории.

Если пользователь нажимает на название темы, то он переходит к списку сообщений, принадлежащих данной теме. В открывшейся странице рядом с названием темы появится ссылка «изменить», которая ведет к странице редактирования темы.

Кнопка ✕ служит для удаления категории. Кнопка ⏻ служит для быстрой публикации категории (или, наоборот, блокирования). При удалении категории удаляются все вложенные подкатегории и сообщения. Категории выводятся в порядке создания.

Редактировать

Название

Показывать на сайте ☒

Удалено модератором ☐

Закрепить тему (всегда сверху) ☐

Закрыть тему ☐

Дата ?

Вложенность: принадлежит

Количество просмотров

Категории имеют следующие характеристики.

Основные:

- **Название.**
- **Опубликовать на сайте.**
- **Закрепить тему (всегда сверху).**
- **Закрыть тему** – Только для тем.
- **Автор.**
- **Дата** – Вводится в формате дд.мм.гггг чч:мм.
- **Редакция** – Пользователь, отредактировавший тему и дата последней редакции.
- **Вложенность: принадлежит** – Категория верхнего уровня, только для разделов и тем. Темы можно переместить только в другой раздел, разделы только в другой блок.
- **Количество просмотров** – Только для разделов и тем.
- **Псевдоссылка** – ЧПУ (человеко-понятные урл url), адрес страницы вида: *http://site.ru/pseudossylka/*. Смотрите параметры сайта.
- **Индексирование для карты сайта** – Категория автоматически индексируется для карты сайта *sitemap.xml*.

Сообщения

Новое поступление желтых птиц [изменить](#)

☐	11.04.2013 11:42	В следующем месяце ожидаем...
☐	11.04.2013 11:27	Спасибо! хорошая альтернатива!
☐	11.04.2013 11:12	А Вы цаплю не рассматривали?...
☐	11.04.2013 11:07	Жалко, что лебедь такой...
☐	11.04.2013 10:58	У коршуна 3 цены в зависимости...
☐	11.04.2013 10:50	А ценовой разброс какой?...
☐	11.04.2013 10:33	Мы привезли 3 новых желтых...

Кнопка ✕ служит для удаления сообщений. Кнопка ⏻ служит для быстрой публикации сообщений (или, наоборот, блокирования).

Редактировать

Показывать на сайте ☒

Удалено модератором ☐

Автор [Филимон Авруцкий\(filimon\)](#)

Дата ?

Тема

Сообщение

Сообщения имеют следующие характеристики.

Основные:

- **Опубликовать на сайте** – Если не отмечено, то сообщение на сайте не отображается.
- **Автор** – Пользователь, добавивший сообщение.
- **Дата** – Вводится в формате дд.мм.гггг чч:мм.
- **Редакция** – Пользователь, отредактировавший сообщение и дата последней редакции.
- **Тема.**
- **Сообщение.**

- Прикрепленные файлы.

Настройки

Настройки

 **Изменения сохранены!**

Количество тем на странице	30
Формат даты	вчера 15:30 ▼
Максимальная вложенность	7
Максимальное количество сообщений на странице	50
Сколько дней хранить "новые" сообщения	3 ?

Использовать защитный код (капчу) ☐

- **Количество тем на странице** – Количество одновременно выводимых тем в списке.
- **Формат даты** – Позволяет настроить отображение даты в модуле. Возможные значения:
 - 01.05.2013,
 - 01.05.2013 14:45,
 - 1 мая 2013 г.,
 - 1 мая,
 - 1 мая 2013, понедельник,
 - вчера 15:30,
 - не отображать.
- **Максимальная вложенность** – Ограничивает вложенность дерева сообщений.
- **Количество сообщений на странице** – Количество одновременно выводимых в сообщений верхнего уровня.
- **Сколько дней хранить «новые» сообщения** – Для чистки мусора в логе новых сообщений. При большом количестве пользователей рекомендуется устанавливать не более трех дней.
- **Использовать ручную сортировку разделов.**
- **Использовать защитный код (капчу)** – Для добавления сообщения пользователь должен ввести защитный код.
- **Только для зарегистрированных пользователей.**
- **Предмодерация темы для обсуждения** – Добавленные темы отображаются на сайте только после одобрения модератором.
- **Предмодерация сообщений** – Добавленные сообщения отображаются на сайте только после одобрения модератором.
- **Разрешить добавление файлов** – Позволяет пользователям прикреплять файлы к сообщениям.
- **Максимальное количество добавляемых файлов** – Количество добавляемых файлов. Если значение равно нулю, то форма добавления файлов не выводится.
- **Доступные типы файлов (через запятую).**

Пример:

doc, gif, jpg, mpg, pdf, png, txt, zip

- **Распознавать изображения** – Позволяет прикрепленные к вопросу файлы в формате JPEG, GIF, PNG отображать как изображения.
- **Размер для большого изображения** – Размер изображения, отображаемый в пользовательской части сайта при увеличении изображения предпросмотра.
- **Размер для маленького изображения** – Размер изображения предпросмотра.
- **Использовать анимацию при увеличении изображений** – Параметр добавляет JavaScript код, позволяющий включить анимацию при увеличении изображений. Параметр выводится, если отмечена опция «Распознавать изображения».
- **Максимальный размер загружаемых файлов** – Параметр показывает максимально допустимый размер загружаемых файлов, установленный в настройках хостинга.

Шаблонные теги

Для работы с модулем «Форум» служат следующие [шаблонные теги](#):

show_block – выводит последние темы.

Атрибуты:

— **count** – количество выводимых тем (по умолчанию 3);

— **cat_id** – категория или раздел форума (id категории, по умолчанию выводятся все темы без учета категории);

— **template** – шаблон тега (файл *modules/forum/views/forum.view.show_block_template.php*; по умолчанию шаблон *modules/forum/views/forum.view.show_block.php*).

Пример:

```
<insert name="show_block_rel" module="forum">
```

выведет 3 похожие на текущую темы

show_block_messages – выводит блок сообщений.

Атрибуты:

— **count** – количество выводимых сообщений;

— **cat_id** – раздел форума;

— **template** – шаблон тега (файл *modules/forum/views/forum.view.show_block_messages_template.php*; по умолчанию шаблон *modules/forum/views/forum.view.show_block_messages.php*).

Пример:

```
<insert name="show_block_messages" module="forum">
```

show_block_rel – на странице темы форума выводит блок похожих тем. Похожие темы подбираются по совпадению слов в заголовках.

Атрибуты:

- **count** – количество выводимых тем;
- **cat_id** – раздел форума;
- **template** – шаблон тега (файл *modules/forum/views/forum.view.show_block_rel_template.php*; по умолчанию шаблон *modules/forum/views/forum.view.show_block_rel.php*).

Пример:

```
<insert name="show_block_rel" module="forum">
```

выведет 3 похожие на текущую темы

Файлы

1. *modules/forum/admin/forum.admin.php* – администрирование модуля;
2. *modules/forum/admin/forum.admin.category.php* – редактирование категорий форума;
3. *modules/forum/admin/forum.admin.config.php* – настройки модуля;
4. *modules/forum/admin/forum.admin.element.php* – редактирование сообщений;
5. *modules/forum/admin/forum.admin.role.php* – права пользователя для пользовательской части, используемые модулем;
6. *modules/forum/forum.php* – контроллер;
7. *modules/forum/forum.ajax.php* – обработка Ajax-запроса на добавление и редактирование тем;
8. *modules/forum/forum.install.php* – установка модуля;
9. *modules/forum/forum.js* – JS-сценарий модуля;
10. *modules/forum/forum.model.php* – модель;
11. *modules/forum/forum.sitemap.php* – карта ссылок для модуля «Карта сайта»;
12. *modules/forum/img/** – изображения;
13. *modules/forum/message/message.ajax.php* – обработка Ajax-запроса на добавление и редактирование сообщений;

14. *modules/forum/message/message.inc.php* – работа с сообщениями в модулей «Форум»;
15. *modules/forum/message/views/message.view.author.php* – шаблон вывода информации о пользователе;
16. *modules/forum/message/views/message.view.edit.php* – шаблон формы редактирования сообщения;
17. *modules/forum/message/views/message.view.form.php* – шаблон формы добавления сообщения;
18. *modules/forum/message/views/message.view.get.php* – шаблон списка и формы добавления сообщений;
19. *modules/forum/message/views/message.view.get_attachments.php* – шаблон прикрепленных к сообщению файлов;
20. *modules/forum/message/views/message.view.id.php* – шаблон сообщения с формой ответа на него и списком ответов;
21. *modules/forum/message/views/message.view.id_message.php* – шаблон сообщения;
22. *modules/forum/message/views/message.view.list.php* – шаблон списка сообщений;
23. *modules/forum/views/forum.view.author.php* – шаблон вывода информации о пользователе;
24. *modules/forum/views/forum.view.edit.php* – шаблон формы редактирования/добавления категории;
25. *modules/forum/views/forum.view.first_page.php* – шаблон первой страницы модуля;
26. *modules/forum/views/forum.view.form_search.php* – шаблон формы поиска по темам и сообщениям;
27. *modules/forum/views/forum.view.id.php* – шаблон страницы темы;
28. *modules/forum/views/forum.view.list_category.php* – шаблон списка категорий форума;
29. *modules/forum/views/forum.view.list_id_category.php* – шаблон отдельной темы в категории;
30. *modules/forum/views/forum.view.list_new.php* – шаблон страницы новых сообщений;
31. *modules/forum/views/forum.view.list_search.php* – шаблон списка найденных сообщений;
32. *modules/forum/views/forum.view.show_block.php* – шаблон блока тем форума;
33. *modules/forum/views/forum.view.show_block_messages.php* – шаблон блока сообщений форума;
34. *modules/forum/views/forum.view.show_block_rel.php* – шаблон блока похожих тем форума.

Опросы

- [Вопросы](#)
- [Ответы](#)
- [Варианты пользователей](#)
- [Настройки](#)
- [Шаблонные теги](#)
- [Файлы](#)

Модуль относится к части CMS «Интерактив» и может быть установлен опционально.

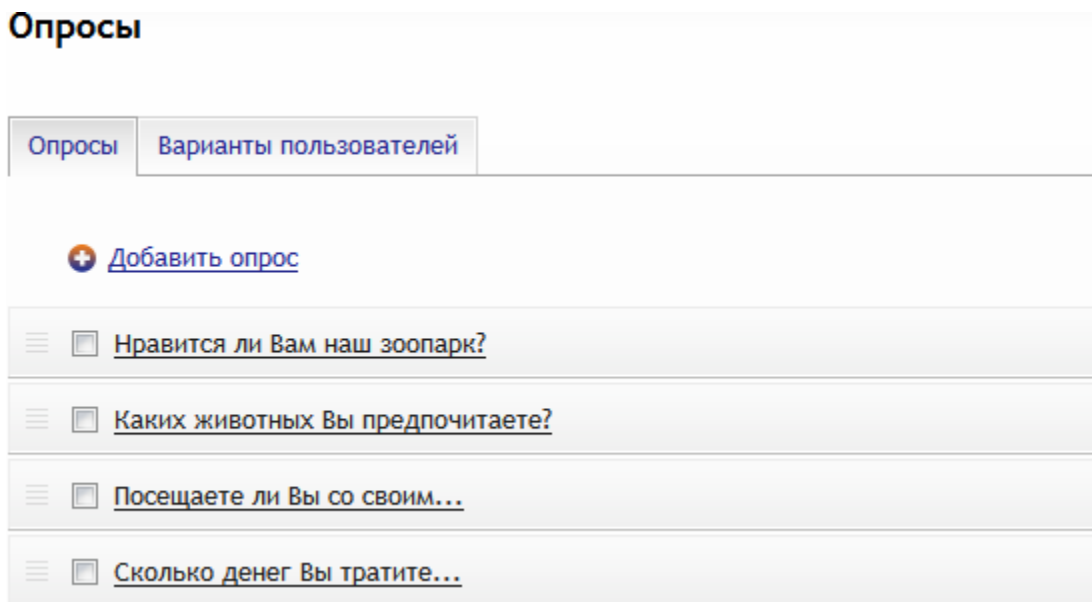
Модуль «Опросы» выводится с помощью шаблонной функции [show_block](#).

Голосование проходит с помощью технологии Ajax, то есть без перезагрузки страницы.

Административная часть модуля состоит из трех частей: вопросы, ответы и варианты пользователей. При открытии модуля выводится список вопросов. При клике на вопрос откроется список ответов на вопрос.

Для редактирования вопроса необходимо нажать на ссылку «изменить» рядом с названием вопроса.

Вопросы



Кнопка ✖ служит для удаления вопросов и ответов. При удалении вопроса удаляются и ответы на него. Кнопка ⏻ служит для быстрой публикации вопроса (или, наоборот, блокирования).

Порядок отображения вопросов и ответов можно редактировать с помощью мыши и кнопки .

Редактировать

Вопрос

Показывать на сайте ☒

Запретить показывать результаты голосования на сайте ☐

Пользователи могут дать свой вариант ответа ☐ ?

Сортировка: установить перед Нравится ли Вам наш зоопарк?

Раздел сайта

- Главная страница
- О нас
- Животные
- Миссия

Вопросы имеют следующие характеристики.

Основные:

- **Вопрос** – Параметр имеет разные значения для разных языковых версий сайта.
- **Опубликовать на сайте** – Публикация на сайте, активность. Параметр имеет разные значения для разных языковых версий сайта.
- **Запретить показывать результаты голосования на сайте** – Пользователи не смогут увидеть результаты голосования.
- **Пользователи могут дать свой вариант ответа** – Позволяет пользователям добавлять свои варианты ответа.
- **Сортировка: установить перед** – Редактирование порядка отображения пункта. Поле доступно для редактирования только для незаблокированных опросов.
- **Раздел сайта** – Позволяет выбрать разделы сайта, на которых будет выводиться опрос.

Ответы

Нравится ли Вам наш зоопарк? [изменить](#)

Опросы **Варианты пользователей**

Категория:

[+ Добавить вариант ответа](#)

- ☐ Сойдет
- ☐ Мне нейтрально

Ответы имеют следующие характеристики.

The screenshot shows a form titled "Редактировать" (Edit) for a survey response. It contains the following fields and controls:

- Ответ** (Answer): A text input field containing the word "Сойдет".
- Показывать на сайте** (Show on site): A checkbox that is checked.
- Количество ответов** (Number of answers): A text input field with a question mark.
- Вопрос** (Question): A text input field containing the question "Нравится ли Вам наш зоопарк?".
- Сортировка: установить перед** (Sort: set before): A text input field containing the word "Сойдет".

At the bottom of the form, there are four buttons: "Сохранить" (Save), "Сохранить и выйти" (Save and exit), "Сохранить и добавить ещё" (Save and add more), and a red button with a white "В" (Back).

Основные:

- **Ответ** – Параметр имеет разные значения для разных языковых версий сайта.
- **Опубликовать на сайте** – Публикация на сайте, активность. Параметр имеет разные значения для разных языковых версий сайта.
- **Количество голосов** – Количество выбравших данный ответ.
- **Вопрос.**
- **Сортировка: установить перед** – Редактирование порядка отображения пункта. Поле доступно для редактирования только для показанных на сайте ответов.

Варианты пользователей

The screenshot shows a form titled "Варианты пользователей" (User Variants). It contains the following fields and controls:

- Опросы** (Surveys) and **Варианты пользователей** (User Variants): Two tabs, with "Варианты пользователей" currently selected.
- Опрос:** (Survey): A dropdown menu showing "Все" (All).
- Птицы** (Birds): A checkbox that is checked.
- Отметить все** (Mark all): A checkbox that is checked.
- Удалить** (Delete): A dropdown menu.
- Применить** (Apply): A blue button.
- На странице:** (On page): A text input field showing "30".
- Изменить** (Change): A blue button.

Варианты пользователей имеют следующие характеристики.

Редактировать

Ответ

Сохранить

Сохранить и выйти

Сохранить и добавить ещё

В

Основные:

- **Ответ** – Ответ пользователя.

Настройки

Настройки

Только для зарегистрированных пользователей ☐ ?

Защита от накруток

Сохранить

- **Только для зарегистрированных пользователей** – Если отмечена, голосовать смогут только зарегистрированные пользователи.
- **Защита от накруток** – Возможные значения:
 - нет,
 - использовать защитный код,
 - вести лог голосовавших,
 - запрещать голосовать повторно.
- **Сортировать ответы по количеству голосов.**

Шаблонные теги

Для работы с модулем «Опросы» служат следующие [шаблонные теги](#):

show_block – выводит опросы.

Атрибуты:

- **id** – задает номер вопроса;
- **count** – количество вопросов. Значение all выведет все вопросы. По умолчанию 1;
- **template** – шаблон тега (файл `modules/votes/views/votes.view.show_block_template.php`; по умолчанию шаблон `modules/votes/views/votes.view.show_block.php`).

Пример:

```
<insert name="show_block" module="votes">
```

выведет вопрос, варианты ответа и кнопки «Ответить» и «Результаты», согласно настройкам и заполнению модуля «Опросы» в административной части сайта

Файлы

1. `modules/votes/admin/votes.admin.php` – администрирование модуля;
2. `modules/votes/admin/votes.admin.category.php` – редактирование вопросов для голосования;
3. `modules/votes/admin/votes.admin.config.php` – настройки модуля;
4. `modules/votes/admin/votes.admin.element.php` – редактирование ответов для голосования;
5. `modules/votes/admin/votes.admin.userversion.php` – список ответов пользователей;
6. `modules/votes/views/votes.view.answers.php` – шаблон результатов голосования;
7. `modules/votes/views/votes.view.form.php` – шаблон вопросов для голосования;
8. `modules/votes/views/votes.view.show_block.php` – шаблон блока опросов;
9. `modules/votes/votes.php` – контроллер;
10. `modules/votes/votes.ajax.php` – обработка запроса при голосовании;
11. `modules/votes/votes.install.php` – установка модуля;
12. `modules/votes/votes.js` – JS-сценарий модуля;
13. `modules/votes/votes.model.php` – модель.

Рассылки

- [Редактирование подписки на сайте](#)
- [E-mail рассылки](#)
- [Категории](#)
- [Подписчики](#)
- [SMS-рассылки](#)
- [Номера телефонов](#)
- [Настройки](#)
- [Шаблонные теги](#)
- [Файлы](#)

Модуль относится к части CMS «Интерактив» и может быть установлен опционально.

Редактирование подписки на сайте

Если модуль подключен к странице сайта, то на этой странице можно отключиться от рассылки и редактировать список категорий, на которые подписчик подключен (если в настройках модуля отмечена опция «Использовать категории»). Форма редактирования подписки доступна для пользователя по специальному хэшу в ссылке, которую пользователь получит на свой электронный адрес. Если адреса нет в базе данных, то форма доступна для этого адреса и без хэша. Редактировать подписку для электронного адреса можно независимо от того, авторизован пользователь на сайте или нет.

Список подписчиков пополняется следующим образом:

- администратором в разделе «Подписчики»;
- при регистрации пользователя;
- при оформлении заказа;
- электронный адрес добавлен из формы на сайте, которая выводится тегом [show_form](#).

При регистрации и оформлении заказа электронный адрес добавляется в базу автоматически или при отмеченной галке «Подписаться на новости», если это задано в настройках модуля.

Зарегистрированный пользователь в настройках аккаунта данных увидит ссылку на редактирование подписки (если есть страница с прикрепленным модулем «Рассылки») или опцию «Подписаться на новости» (если она подключена в настройках модуля).

Е-mail рассылки

Рассылки [/ изменить](#)

[Рассылки](#) [Категории](#) [Подписчики](#) [SMS-рассылки](#) [Номера телефонов](#)

Категория:

Название:

 [Добавить рассылку](#)

☐ ТВ	11.04.2013	ТВ-ролик
☐ Новости сайта	10.04.2013	Сезон скидок!

Выпуски рассылки выводятся в порядке отправки. Кнопка  служит для удаления выпуска.

Редактировать

Название

Дата добавления Thu, 11 Apr 2013 11:57:17?

Отправить ☐

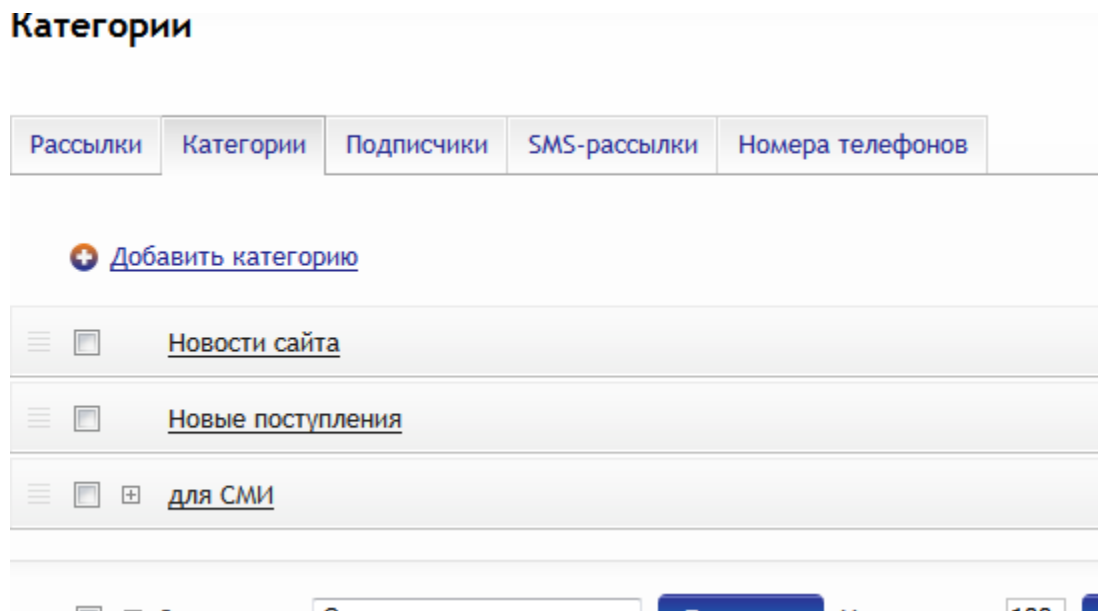
Описание

HTML |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |

Рассылка отправляется только в том случае, если заполнены поля «Описание», «Отправить» и рассылка не была отправлена ранее.

Рассылка отправляется по электронным адресам из раздела «Подписчики».

Категории



Категории рассылки могут иметь неограниченное число подкатегорий. При необходимости категории можно не использовать. Для этого нужно отключить параметр «Использовать категории» в настройках модуля.

Выпуск рассылки можно отнести к нескольким категориям, чтобы охватить пользователей, подписанных на разные категории.

Добавить подкатеорию к категории можно нажав на знак  рядом с названием категории. Порядок отображения категорий можно редактировать с помощью мыши и кнопки . Кнопка  служит для удаления категории. Кнопка  служит для быстрой публикации категории (или, наоборот, блокирования). При удалении категории удаляются все вложенные подкатегории и выпуски рассылок.

[Редактировать](#)

Основные:

- **Название** – Параметр имеет разные значения для разных языковых версий сайта.
- **Опубликовать на сайте.**
- **Сортировка: установить перед** – Редактирование порядка отображения пункта. Поле доступно для редактирования только для незаблокированных категорий.
- **Вложенность: принадлежит** – Категория верхнего уровня.
- **Описание** – Описание категории. Параметр имеет разные значения для разных языковых версий сайта.

Подписчики

Подписчики

Рассылки

Категории

Подписчики

SMS-рассылки

Номера телефонов

 [Добавить подписчика](#)

☐

test_diafan@mail.ru

☐ Отметить все

Сделать активным 

Применить

На странице: 30

Изменить

Редактировать

Почтовый ящик

Дата добавления ?

Получает рассылку ☒

Категории рассылок

- ☒ Новости сайта
- ☒ Новые поступления
- ☒ для СМИ
- ☒ Газеты
- ☒ ТВ

Основные:

- **Почтовый ящик** – Электронный адрес подписчика.
- **Имя получателя** – В рассылке не участвует.
- **Дата добавления** – Дата добавления подписчика. Вводится в формате дд.мм.гггг чч:мм.
- **Получает рассылку** – Позволяет отключить подписчика от рассылки. Пользователи, отписавшиеся от рассылки на сайте из базы данных не удаляются, только снимается отметка «Получает рассылку».
- **Код управления рассылкой** – Код, указанный пользователю в ссылке на управление рассылкой.

SMS-рассылки

SMS-рассылки

[Рассылки](#)
[Категории](#)
[Подписчики](#)
[SMS-рассылки](#)
[Номера телефонов](#)

Название: [Найти](#)

[+ Добавить SMS-рассылку](#)

☐	12.04.2013	Рассылка для клиентов
--------------------------	------------	---------------------------------------

☐ Отметить все
 [Удалить](#)
[Применить](#)
 На странице:
[Изменить](#)

SMS-рассылка возможна, если в [«Параметрах сайта»](#) подключена опция «Подключить SMS-уведомления». Выпуски SMS-рассылки выводятся в порядке отправки. Кнопка ✖ служит для удаления выпуска.

Редактировать

Название

Рассылка для клиентов

Дата добавления

Fri, 12 Apr 2013 12:47:04?

Отправить

☐

Текст рассылки

Скидки!!!
Цена на билеты в наш зоопарк в выходные
снижена в 2 раза!

Выпуски SMS-рассылки имеют следующие характеристики.

Основные:

- **Название** – Наименование рассылки. В SMS не подставляется. Используется администратором для удобства.
- **Дата добавления** – Отображается дата добавления или отправления рассылки.
- **Отправить** – Если отметить эту галку и сохранить, рассылка начнет отправляться. Если не отмечать галку, рассылка будет сохранена как черновик. Рассылка отправляется один раз, если отмечено поле «Отправить».
- **Текст рассылки** – Текст SMS-сообщения латинскими буквами.

Номера телефонов

Номера телефонов

Рассылки

Категории

Подписчики

SMS-рассылки

Номера телефонов

 [Добавить номер телефона](#)

☐

+7922555555

☐ Отметить все

Сделать активным

Применить

На странице: 30

Изменить

Раздел содержит номера телефонов, по которым происходит SMS-рассылка. Рассылка идет только по номерам, помеченным опцией «Получает рассылку». База номеров заполняется вручную из панели администрирования.

Редактировать

Номер телефона в федеральном формате

+7922555555

Дата добавления

12.04.2013

 ?

Получает рассылку

☒

Сохранить

Сохранить и выйти

Сохранить и добавить ещё

В

Подписчики имеют следующие характеристики.

Основные:

- **Номер телефона в федеральном формате** – 11 цифр номера мобильного телефона.
- **Имя** – Имя получателя. В рассылке не участвует.
- **Дата добавления** – дата добавления номера в базу данных.
- **Получает рассылку** – Позволяет отключить телефон от рассылки.

Настройки

Настройки

Использовать категории

☒

Тема письма для рассылки

Рассылка сайта %title (%url). %subject

E-mail, указываемый в обратном адресе пользователю

test_diafan@mail.ru

Выводить при регистрации галку "Подписаться на новости"

☐

Выводить при оформлении заказа галку "Подписаться на новости"

☐

Сообщение после добавления e-mail

E-mail успешно добавлен. Вам отправл

- **Использовать категории** – Позволяет включить/отключить категории рассылки.
- **Тема письма для рассылки** – Можно добавлять:
 - %title – название сайта,
 - %url – адрес сайта (например, site.ru),

- %subject – тема рассылки.

Параметр имеет разные значения для разных языковых версий сайта.

Пример:

Рассылка сайта %title (%url). %subject

- **Шаблон письма для рассылки** – Можно добавлять:

- %title – название сайта,
- %url – адрес сайта (например, site.ru),
- %text – текст рассылки,
- %link – ссылка для редактирования категорий рассылки,
- %actlink – ссылка для отмены рассылки.

Параметр имеет разные значения для разных языковых версий сайта.

Пример:

Рассылка сайта %title (%url).
 %text Для изменения списка категорий рассылок
пройдите по ссылке %link.
 Если Вы хотите отписаться от
рассылки, перейдите по ссылке %actlink.

- **E-mail, указываемый в обратном адресе пользователю** – Возможные значения:
 - e-mail, указанный в параметрах сайта;
 - другой (при выборе этого значения появляется дополнительное поле **впишите e-mail**).

Параметр имеет разные значения для разных языковых версий сайта.

- **Выводить при регистрации галку «Подписаться на новости»** – При регистрации пользователь может подписаться на рассылку. Если опция отключена, пользователь будет подписан автоматически.
- **Выводить при оформлении заказа галку «Подписаться на новости»** – При оформлении заказа пользователь может подписаться на рассылку. Если опция отключена, пользователь будет подписан автоматически.
- **Сообщение после добавления e-mail** – Сообщение пользователю, после успешной подписки на рассылку. Параметр имеет разные значения для разных языковых версий сайта.

Пример:

E-mail успешно добавлен. Вам отправлено уведомление.

- **Тема письма для уведомлений пользователя о подписке на рассылку** – Тема письма, отправляемого пользователю, после успешной подписки на рассылку. Можно добавлять:
 - %title – название сайта,
 - %url – адрес сайта (например, site.ru).

Параметр имеет разные значения для разных языковых версий сайта.

Пример:

Подписка на рассылку с сайта %title (%url)

- **Сообщение для уведомлений пользователя о подписке на рассылку** – Текст письма, отправляемого пользователю, после успешной подписки на рассылку. Можно добавлять:

- %title – название сайта,
- %url – адрес сайта (например, site.ru),
- %subject – тема рассылки,
- %link – ссылка для редактирование категорий рассылки, на которые подписан пользователь,
- %actlink – ссылка, по которой подписчик будет отключен от рассылки.

Параметр имеет разные значения для разных языковых версий сайта.

Пример:

Здравствуйте! Вы подписались на рассылку с сайта %title (%url).
 Для изменения списка категорий рассылок пройдите по ссылке %link.
 Если Вы хотите отписаться от рассылки, пройдите по ссылке %actlink.

Шаблонные теги

Для работы с модулем «Рассылки» служат следующие [шаблонные теги](#):

show_form – выводит форму подписки на рассылки.

Атрибуты:

— **template** – шаблон тега (файл *modules/subscription/views/subscription.view.form_***template***.php*; по умолчанию шаблон *modules/subscription/views/subscription.view.form.php*).

Пример:

```
<insert name="show_form" module="subscription">
```

выведет форму подписки на рассылки

Файлы

1. *modules/subscription/admin/subscription.admin.php* – рассылки;
2. *modules/subscription/admin/subscription.admin.category.php* – редактирование категорий рассылки;
3. *modules/subscription/admin/subscription.admin.config.php* – настройки модуля;
4. *modules/subscription/admin/subscription.admin.emails.php* – база электронных ящиков для рассылок;
5. *modules/subscription/admin/subscription.admin.phones.php* – база номеров телефонов для SMS рассылки;
6. *modules/subscription/admin/subscription.admin.sms.php* – рассылки по SMS;
7. *modules/subscription/subscription.php* – контроллер;

8. *modules/subscription/subscription.ajax.php* – обработка запроса при отправке сообщения из формы подписки на рассылку;
9. *modules/subscription/subscription.export.php* – экспорт телефонов и электронных ящиков;
10. *modules/subscription/subscription.install.php* – установка модуля;
11. *modules/subscription/subscription.model.php* – модель;
12. *modules/subscription/views/subscription.view.edit.php* – шаблон форма редактирование подписки на новости;
13. *modules/subscription/views/subscription.view.error.php* – шаблон информационного сообщения;
14. *modules/subscription/views/subscription.view.form.php* – шаблон формы подписки на рассылки.

Карта сайта

- [Файлы](#)

Модуль генерирует карту сайта. В карту сайта входят все активные страницы сайта, не отмеченные параметром «Не показывать на карте сайта», а также прикрепленные к страницам сайта активные категории модулей, не помеченные параметром «Не показывать на карте сайта».

Модуль имеет только пользовательскую часть и является обязательным для установки.

diafan.CMS генерирует также карту сайта для поисковых систем: ***<http://site.ru/sitemap.xml>***.

Файл *sitemap.xml* генерируется на лету, физически файл не создается. Он содержит все ссылки, создаваемые системой, кроме ссылок на элементы, помеченные параметром «Не показывать на карте сайта». Для генерирования ссылок модуля необходимо наличие файла *modules/модуль/модуль.sitemap.php*. Формат файла:

Пример:

```
<?php
if (! defined('DIAFAN'))
{
    include dirname(dirname(dirname(__FILE__))).'/includes/404.php';
}

class Модуль_sitemap extends Diafan
{
    /**
     * Возвращает настройки для генерирования карты модуля
     *
     * @param integer $site_id номер страницы сайта
     * @return array
     */
    public function config($site_id)
    {
        $result = array(
            // тип данных – элемент, категория
            'type' => array('element', 'category'),
            // данные могут быть по разному показаны (активны) на разных языковых версиях
            'multilang' => true,
            // условия для выборки
            'where' =>
                array(
                    // для типа «элемент»
                    'element' => "AND map_no_show='0'",
                    // для типа «категория»
                    'category' => "AND map_no_show='0'",
                )
        );
    }
};
```

```
    return $result;  
  }  
}
```

Файлы

1. *modules/map/admin/map.admin.ajax.php* – подгружает карту сайта в визуальном редакторе;
2. *modules/map/admin/map.admin.inc.php* – подключение модуля к административной части других модулей;
3. *modules/map/admin/map.tiny.js* – подгружает карту сайта в визуальном редакторе, JS-сценарий;
4. *modules/map/map.php* – контроллер;
5. *modules/map/map.inc.php* – подключение модуля;
6. *modules/map/map.install.php* – установка модуля;
7. *modules/map/map.model.php* – модель;
8. *modules/map/map.sitemap.php* – карта сайта в XML формате;
9. *modules/map/views/map.view.list.php* – шаблон списка страниц в карте сайте.

Ошибка на сайте

- [Шаблонные теги](#)
- [Файлы](#)

Модуль «Ошибка на сайте» позволяет получать сообщения от пользователей об ошибках в текстах сайта.

Модуль относится к части CMS «Интерактив» и может быть установлен опционально.

Ошибки на сайте

☐	11.04.2013 11:57	/shop/waterfowl/beaver/	5-пальми	П: не
--------------------------	------------------	---	----------	----------

☐ Отметить все

Удалить ▾

Применить

На странице: 30

Изменить

На сайте модуль подключается с помощью тега [show_block](#). Тег выводит надпись «Если Вы заметили ошибку на сайте, выделите ее и нажмите Ctrl+Enter» и код для обработки нажатия сочетания клавиш Ctrl+Enter.

В административной части модуля агрегируются сообщения пользователей об ошибках на сайте. Количество сообщений (не нулевое) выведено в меню административной части рядом с названием модуля.

Редактировать

Дата и время

URL

Выделенный текст

Комментарий

Сообщения имеют следующие характеристики:

Основные:

- **Дата и время** – Дата и время добавления сообщения.
- **URL** – Относительный адрес страницы, на которой ошибка найдена.
- **Выделенный текст** – Фрагмента текста на сайте, который пользователь выделили перед тем, как отправить сообщение.
- **Комментарий** – Комментарий, оставленный пользователем, добавившим сообщение.

Шаблонные теги

Для работы с модулем «Ошибка на сайте» служат следующие [шаблонные теги](#):

show_block – подключает модуль «Ошибка на сайте».

Пример:

```
<insert name="show_block" module="mistakes">
```

Выведет «Если Вы заметили ошибку на сайте, выделите ее и нажмите Ctrl+Enter».

Шаблон: файл *modules/mistakes/views/mistakes.show_block.php*.

Файлы

1. *modules/mistakes/admin/mistakes.admin.php* – просмотр уведомлений об ошибках на сайте;
2. *modules/mistakes/admin/mistakes.admin.count.php* – количество уведомлений об ошибках на сайте;

3. *modules/mistakes/mistakes.php* – контроллер;
4. *modules/mistakes/mistakes.ajax.php* – обрабатывает полученные данные из формы;
5. *modules/mistakes/mistakes.install.php* – установка модуля;
6. *modules/mistakes/mistakes.js* – JS-сценарий модуля;
7. *modules/mistakes/views/mistakes.view.show_block.php* – шаблон формы добавления уведомления об ошибке.

On-line консультант

- [Шаблонные теги](#)
- [Файлы](#)

Модуль «On-line консультант» подключает сайт к системе [RedHelper](#) и позволяет установить на сайте форму для связи пользователя с консультантом в режиме on-line.

On-line консультант



Вы зарегистрированы в системе Red Helper.

Ваш логин и пароль: test_diafan и admin.

[Личный кабинет на сайте redhelper.ru](#)

Для начала работы с консультантом нужно проделать три шага:

1. Добавьте в шаблон diafan.CMS (/themes/site.php) шаблонный тег `<insert name="show_block" module="consultant">`.
2. [Скачайте](#) и установите приложение оператора. Это независимая программа, устанавливающая позволяет наблюдать за посетителями, отвечать на их вопросы и многое другое. Для других операционных систем и мобильных устройств Вы можете [использовать любой](#) : приложение используйте логин и пароль, обозначенные выше.

Модуль относится к части CMS «Интерактив» и может быть установлен опционально.



Онлайн консультант на сайт – мощное решение, дающее возможность мгновенно общаться с посетителями.

Основные преимущества работы Онлайн консультанта RedHelper:

1. Совместный браузер с возможностью увидеть сайт глазами ваших клиентов;
2. Код системы не влияет на скорость загрузки сайта;
3. Позволяет полностью настроить внешний вид и поведение системы на сайте с помощью API;
4. Полный контроль за работой ваших операторов;
5. Высокая стабильность и надежность в работе;
6. Настройка отделов.

Для пользователей RedHelper предлагаются следующие условия:

1. Быстрая и удобная настройка системы на сайте;
2. Лучшая техническая поддержка и персональный менеджер, который обучит ваших операторов и самостоятельно настроит систему;
3. Тестовый период обкатки системы с полным функционалом;
4. Возможность установки единого консультанта на все используемые вами домены.



On-line консультант



Онлайн консультант на сайт - мощное решение, дающее воз посетителями.

Поднимает продажи и создает качественный сервис.
Просто и очень эффективно.

Смотрите демонстрацию возможностей:



Административная часть модуля нужна для регистрации в системе RedHelper и настройки внешнего вида кнопки и формы на сайте.

On-line консультант

Логин:

Пароль:

Цвет кнопки:

Расположение:

Сокращенный вид: ☐

Положение чата по горизонтали:

Положение чата по вертикали:

Верхний текст с названием компании:

Если у вас уже есть аккаунт в системе, то сразу переходите в настройки модуля по ссылке [внизу](#).

В настройках можно указать:

- **Логин** – обязательно для работы формы.
- **Пароль** – не обязательно для заполнения.
- **Цвет кнопки** – описание цвета в [формате, принятом для HTML](#).
- **Расположение** – Справа, слева или сверху сайта. По умолчанию – слева.
- **Сокращенный вид** – Кнопка уже, чем стандартная. По умолчанию – отключено.
- **Положение чата по горизонтали** – В пикселях. По умолчанию – 30.
- **Положение чата по вертикали** – В пикселях. По умолчанию – 50.
- **Верхний текст с названием компании** – По умолчанию – *Наша компания*.
- **Текст под названием компании** – По умолчанию – *Лучшие товары*.
- **Текст приглашения** – По умолчанию – *Вам чем-нибудь помочь?*.
- **Время задержки на выдачу приглашения** – По умолчанию – 20.
- **Ширина окна чата** – В пикселях. По умолчанию – 200.
- **Высота окна чата** – В пикселях. По умолчанию – 300.

На сайте on-line консультант подключается с помощью тега [show_block](#).

Для работы нужно [скачать](#) и установить приложение оператора. Это независимая программа, устанавливаемая на компьютер (Windows), позволяет наблюдать за посетителями, отвечать на их вопросы и многое другое.

Для других операционных систем и мобильных устройств можно [использовать любой jabber клиент](#).

Шаблонные теги

Для работы с модулем «On-line консультант» служат следующие [шаблонные теги](#):

show_block – подключает on-line консультант.

Атрибуты:

— **template** – шаблон тега (файл *modules/consultant/views/consultant.view.show_block_template.php*; по умолчанию шаблон *modules/consultant/views/consultant.view.show_block.php*).

Пример:

```
<insert name="show_block" module="consultant">
```

выведет код для подключения on-line консультанта

Файлы

1. *modules/consultant/admin/consultant.admin.php* – On-line консультант, система LiveTex;
2. *modules/consultant/admin/consultant.admin.jivosite.php* – On-line консультант, система JivoSite;
3. *modules/consultant/admin/consultant.admin.redhelper.php* – On-line консультант, система RedHelper;
4. *modules/consultant/consultant.php* – контроллер;
5. *modules/consultant/consultant.css* – оформление on-line консультанта системы RedHelper;
6. *modules/consultant/consultant.install.php* – установка модуля;
7. *modules/consultant/consultant.model.php* – модель;
8. *modules/consultant/img/** – изображения;
9. *modules/consultant/views/consultant.view.show_block.php* – шаблон on-line консультанта.

Web Effector

- [Медиапланер](#)
- [Запросы](#)
- [Файлы](#)

Модуль «Web Effector» подключает сайт к системе [Web Effector](#).

Модуль относится к части CMS «Сервис» и может быть установлен опционально.



ВебЭффектор – эффективный и автоматический сервис продвижения сайтов в ТОП поисковых систем. Незаменимый инструмент для владельцев сайтов и вебмастеров, а для SEO-специалистов – лучший друг и помощник в работе.

Для начала работы необходимо зарегистрироваться. Модуль предложит форму регистрации нового пользователя. Если у вас уже есть аккаунт, пройдите в настройки модуля (по ссылке ниже формы регистрации), чтобы авторизоваться в системе.

WebEffector

[Медиапланер](#) [Запросы](#)



ВебЭффектор - эффективный и автоматический сервис продвижения сайтов в ТОП поисковых систем. Незаменимый инструмент для владельцев сайтов и вебмастеров, а для SEO-специалистов – лучший друг и помощник в работе.

Для начала работы требуется дополнительная регистрация в системе Web Effector.

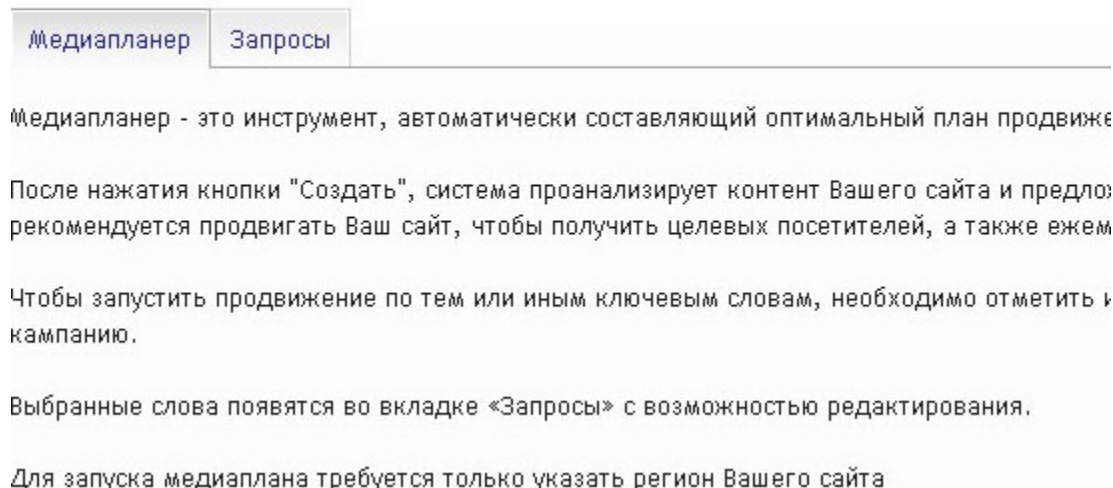
E-mail:

Пароль:

[Зарегистрироваться](#)

После авторизации в системе необходимо создать проект, указав регион продвижения для сайта. Проект создается с названием и адресом текущего сайта.

WebEffector



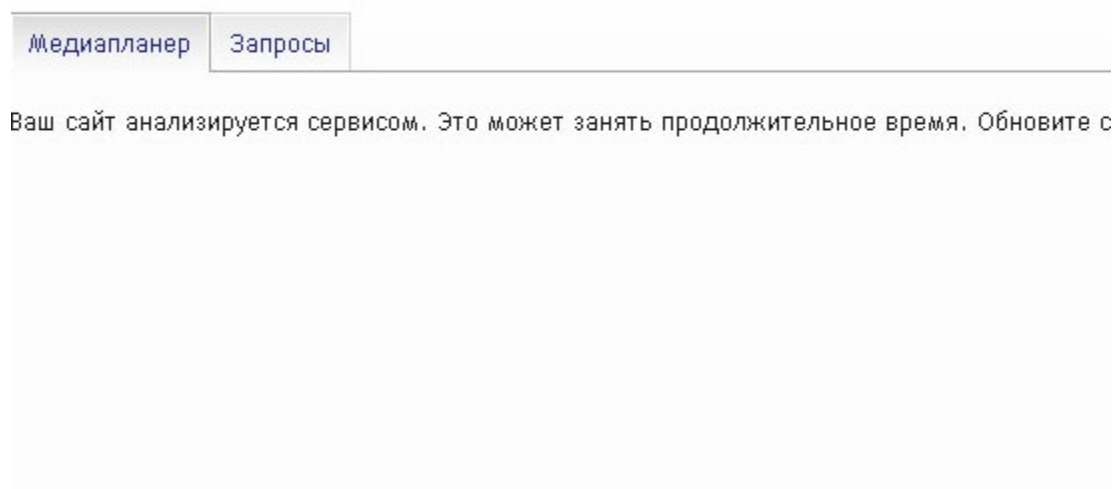
Модуль состоит из двух частей: Медиапланер и Запросы.

Медиапланер

Медиапланер – это инструмент, автоматически составляющий оптимальный план продвижения сайта.

После нажатия кнопки «Создать» система проанализирует контент сайта и предложит ключевые слова, по которым рекомендуется продвигать сайт, чтобы получить целевых посетителей, а также ежемесячный бюджет. Это займет некоторое время, о чем модуль сообщит.

WebEffector



Чтобы запустить продвижение по тем или иным ключевым словам, необходимо отметить их в предложенном списке и добавить в кампанию.

WebEffector

Медиапланер		Запросы
Ключевое слово		Страница
☐	кондиционеры	http://yoursite.ru/
☐	купить кондиционер	http://yoursite.ru/
☐	кондиционеры Daikin	http://yoursite.ru/catalog/
☐	вентиляторы	http://yoursite.ru/catalog/
☐	установка кондиционеров	http://yoursite.ru/uslugi/
☐	обслуживание кондиционеров	http://yoursite.ru/uslugi/
☐	ремонт кондиционеров	http://yoursite.ru/uslugi/remont/
☐	Отметить все	

Запросы

Выбранные слова появятся во вкладке «Запросы». Запросы можно удалить из кампании. Если достаточно средств на счете в системе WebEffecotor, запросы можно запустить в продвижение.

WebEffector

Медиапланер

Запросы

Добавить запрос

Ключевое слово	GEO	ТОП	URL	Бюджет
☐ кондиционеры	нет	10	...yoursite.ru/	0 р.
☐ купить кондиционер	нет	10	...yoursite.ru/	0 р.
☐ кондиционеры Daikin	нет	10	...te.ru/catalog/	0 р.
☐ вентиляторы	нет	10	...te.ru/catalog/	0 р.
☐ установка кондиционеров	нет	10	...ite.ru/uslugi/	0 р.
☐ обслуживание кондиционеров	нет	10	...ite.ru/uslugi/	0 р.
☐ ремонт кондиционеров	нет	10	.../uslugi/remont/	0 р.

После запуска в продвижение система закупает ссылки в рамках бюджета и выводит результаты мониторинга позиций по запросам.

WebEffector

Медиапланер		Запросы		
Добавить запрос				
Ключевое слово	GEO	 	ТОП URL	Бюджет
☐ кондиционеры	да	23 21	10 ...yoursite.ru/	5490 р.
☐ купить кондиционер	да	12 9	10 ...yoursite.ru/	4590 р.
☐ кондиционеры Daikin	да	7 5	10 ...te.ru/catalog/	2480 р.
☐ вентиляторы	да	5 4	10 ...te.ru/catalog/	1650 р.
☐ установка кондиционеров	да	19 11	10 ...ite.ru/uslugi/	3900 р.
☐ обслуживание кондиционеров	да	8 7	10 ...ite.ru/uslugi/	2190 р.
☐ ремонт кондиционеров	да	15 10	10 .../uslugi/remont/	1390 р.

Файлы

1. *modules/webeffector/admin/img/** – изображения для административной части;
2. *modules/webeffector/admin/webeffector.admin.php* – продвижение сайта с помощью системы WebEffector;
3. *modules/webeffector/admin/webeffector.admin.js* – JS-сценарий административной части модуля;
4. *modules/webeffector/admin/webeffector.admin.regions.php* – список регионов для определения региона кампании;
5. *modules/webeffector/webeffector.install.php* – установка модуля.

Файловый менеджер

- [Файлы](#)

Модуль «Файловый менеджер» позволяет работать с файловой системой сайта – редактировать скрипты, загружать и удалять файлы.

Файловый менеджер

Внимание! Все изменения необратимы. Для корректной работы diafan.CMS не рекомендуе userfiles и cache.

 [Создать папку](#)

Укажите путь к файлам

Или перетащите мыш

		 adm
		 cache
		 css

Модуль относится к части CMS «Сервис» и может быть установлен опционально.

Внимание! Модуль должен быть доступен только для веб-мастера! Для этого настройте [права доступа](#) у всех остальных пользователей.

Из-за особой системы доступа к файлам и директориям UNIX-систем для работы с файлами нужно либо поставить права доступа на редактируемые файлы и директории 777 (права на редактирование для всех пользователей), либо настроить доступ по протоколу FTP в «Параметрах сайта».

Для редактирования названия и содержания (для текстовых типов файлов) файлов и директорий нажмите на название файла или директории. В открывшемся окне можно также скачать файл и загрузить файлы в директорию.

Если перед названием директории имеется знак +, то эта директория содержит другие директории и файлы. Создать директорию в другой директории можно, нажав на знак  рядом с названием директории. Кнопка  служит для удаления директории или файла.

Файлы

1. *modules/filemanager/admin/filemanager.admin.php* – файловый менеджер;
2. *modules/filemanager/admin/filemanager.admin.js* – файловый менеджер, JS-сценарий;
3. *modules/filemanager/filemanager.install.php* – установка модуля.

Надмодули

Надмодули – это модули, которые сами по себе отдельным модулем не являются. Они не имеют стандартную структуру модуля diafan.CMS и их можно подключать к другим классическим модулям.

Комментарии

Модуль комментариев подключается к любым элементам на сайте, от новостей и статей, до отдельных позиций магазина и фотогалереи и позволяет сделать сайт по-настоящему интерактивным.

Рейтинг

Как и комментарии, модуль рейтинга можно подключать к любым элементам сайта, от новостей и фотографий до товаров магазина.

Перелинковка

Модуль позволяет задать ссылку, которая будет автоматически подставляться в тексты на сайте с заданным ключевым словом или фразой.

Панель быстрого редактирования

Редактирование данных модулей и интерфейс из пользовательской части сайта.

Изображения

Работа с изображениями, прикрепленными к элементам других модулей.

Прикрепленные файлы

Работа с файлами, прикрепленными к элементам других модулей.

Постраничная навигация

Формирование постраничной навигации контента в других модулях.

Защитный код

Формирование защитного кода (капчи) в формах других модулей.

Комментарии

- [Подключение](#)
- [Как добавлять комментарии к модулю](#)
- [Комментарии](#)
- [Конструктор формы](#)
- [Настройки](#)
- [Шаблонные теги](#)
- [Файлы](#)

Модуль относится к части CMS «Интерактив» и может быть установлен опционально.

Модуль «Комментарии» подключается к другим модулям. В стандартной комплектации diafan.CMS «Комментарии» подключены к следующим модулям: Фотогалерея, Новости, Статьи, Вопрос-Ответ, Магазин, Файлы, Страницы сайта, Объявления. В рамках модуля комментарии подключаются в настройках конкретного модуля.

В пользовательской части сайта комментарии добавляются с помощью технологии Ajax, то есть без перезагрузки всей страницы. Комментарии, содержащие слова длиннее 20 символов, не пропускаются.

Подключение

Подключаемая часть – файл *modules/comments/comments.inc.php*. В нем описан класс *Comments_inc*. В модуле к объекту класса можно обратиться через переменную `$this->diafan->_comments`.

Экземпляр класса создается при первом вызове переменной.

Методы

`string get([integer $element_id = 0], [string $module_name = ''], [integer $site_id = 0], [boolean $is_category = false])` – показывает комментарии, прикрепленные к элементу, и форму добавления комментария.

- access: public
- integer \$element_id: номер элемента модуля
- string \$module_name: название модуля
- integer \$site_id: страница сайта, к которой прикреплен элемент
- boolean \$is_category: это категория

Пример:

```
// получаем комментарии к новости и форму добавления комментария в файле
modules/news/news.model.php
$comments = $this->diafan->_comments->get();
// выводим комментарии в шаблоне модуля в файле
```

```
modules/news/views/news.view.id.php
echo $comments;`
```

Как добавлять комментарии к модулю

В настройках модуля (*modules/модуль/admin/модуль.admin.config.php*) и в файле редактирования элементов модуля (*modules/модуль/admin/модуль.admin.php*) необходимо добавить параметр **'comments'**:

Пример:

```
public $variables = array(
    'main' => array(
        'comments' => 'module',
        ...
    ),
    ...
);
```

Комментарии

Комментарии		
КомментарииКонструктор формы		
☐	11.04.2013 11:57	<u>Мы тоже радуемся вслед за Вами!</u>
☐	11.04.2013 11:57	<u>Вау, какая изворотливая!</u>
☐	10.04.2013 11:57	<u>Ура! Весна! Мы так рады!</u>
☐	10.04.2013 11:57	<u>Отличная статья. Я в восторге. Автор.</u>
☐	09.04.2013 11:57	<u>Красавец</u>
— —		

В административной части модуля все комментарии собираются в одну таблицу и отображаются в порядке поступления. Комментарии можно сортировать по принадлежности к отдельному элементу. Для этого предусмотрены ссылки «Комментарии» в административной части модулей, к которым подключаются комментарии. К примеру, при редактировании фотографии появится ссылка «Комментарии» на комментарии к этой фотографии или надпись «Комментариев нет» при их отсутствии.

Кнопка ✖ служит для удаления комментария. Кнопка ⏻ служит для быстрой публикации комментария (или, наоборот, блокирования).

При удалении какого-либо элемента, удаляются все комментарии, прикрепленные к нему.

Редактировать

Комментарий

Мы тоже радуемся вслед за Вами!

Дата добавления 11.04.2013 11:57 ?

Показывать на сайте ☒ ?

Вложенность: принадлежит [Ура! Весна! Мы так рады!](#)

Объект <http://test.diafan.ru/andrey/news/cat1/show8/>

Комментарии имеют следующие характеристики.

Основные:

- **Дата добавления** – В формате дд.мм.гггг чч:мм.
- **Пользователь** – Пользователь, добавивший комментарий (если комментарий добавлен зарегистрированным пользователем).
- **Комментарий**.
- **Дополнительные поля** – Поля, добавленные в конструкторе формы.
- **Опубликовать на сайте** – Если не отмечена, комментарий не будет виден на сайте.
- **Комментарий к** – Объект, к которому прикреплены комментарии, ссылка на все комментарии к этой странице.
- **Вложенность: принадлежит** – Комментарий верхнего уровня.

Конструктор формы

Конструктор формы

Комментарии Конструктор формы

Модуль:

 [Добавить поле](#)

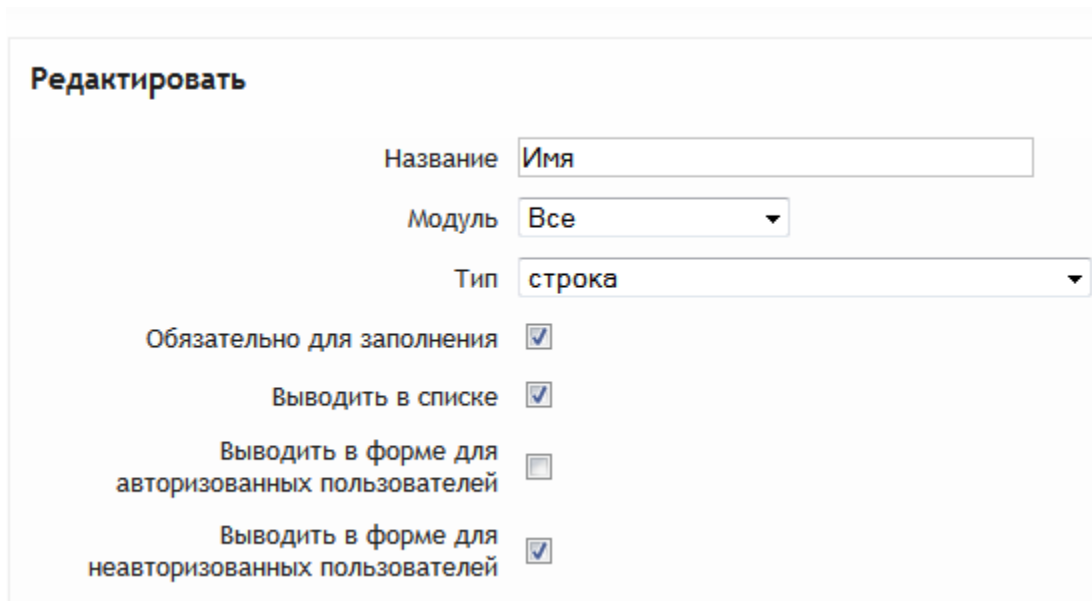
Имя	строка
Имя	строка

Отметить все Удалить Применить На странице: 30 Изменить

Количество и содержание полей комментариев может быть различно. Это обеспечивает конструктор формы.

Порядок отображения полей можно редактировать с помощью мыши и кнопки . Кнопка  служит для удаления полей.

Конструктор – это дополнительные поля формы комментариев.



Дополнительные поля имеют следующие параметры.

Основные:

- **Название** – Параметр имеет разные значения для разных языковых версий сайта.
- **Модуль** – Возможность ограничить применением поля комментариями, прикрепленными к выбранному модулю.
- **Тип** – Возможные значения:
 - строка,
 - число,
 - дата,
 - дата и время,
 - текстовое поле,
 - галочка,
 - выпадающий список,
 - список с выбором нескольких значений,
 - электронный ящик,
 - заголовок группы характеристик,
 - файлы,
 - изображения.
- **Максимальное количество добавляемых файлов** – Количество добавляемых файлов. Если значение равно нулю, то форма добавления файлов не выводится. Параметр выводится, если тип характеристики задан как «файлы».
- **Доступные типы файлов (через запятую)** – Параметр выводится, если тип характеристики задан как «файлы».
- **Распознавать изображения** – Позволяет прикрепленные файлы в формате JPEG,

GIF, PNG отображать как изображения. Параметр выводится, если тип характеристики задан как «файлы».

- **Размер для большого изображения** – Размер изображения, отображаемый в пользовательской части сайта при увеличении изображения предпросмотра. Параметр выводится, если тип характеристики задан как «файлы» и отмечена опция «Распознавать изображения».
- **Размер для маленького изображения** – Размер изображения предпросмотра. Параметр выводится, если тип характеристики задан как «файлы» и отмечена опция «Распознавать изображения».
- **Использовать анимацию при увеличении изображений** – Параметр добавляет JavaScript код, позволяющий включить анимацию при увеличении изображений. Параметр выводится, если отмечена опция «Распознавать изображения». Параметр выводится, если тип характеристики задан как «файлы» и отмечена опция «Распознавать изображения».
- **Максимальный размер загружаемых файлов** – Параметр показывает максимально допустимый размер загружаемых файлов, установленный в настройках хостинга. Параметр выводится, если тип характеристики задан как «файлы».
- **Генерировать размеры изображений** – Размеры изображений, заданные в модуле «Изображения». Параметр выводится, если тип характеристики задан как «изображение».
- **Обязательно для заполнения.**
- **Выводить в списке** – Выводит значение поля для комментариев на сайте в списке комментариев
- **Выводить в форме для авторизованных пользователей.**
- **Выводить в форме для неавторизованных пользователей.**
- **Сортировка: установить перед** – Редактирование порядка следования поля в форме
- **Описание** – Параметр имеет разные значения для разных языковых версий сайта.

Если выбран тип «выпадающий список» или «список с выбором нескольких значений», то появятся дополнительные поля со значениями. Добавить значение можно, нажав  внизу последнего поля, сортировать значения можно с помощью мыши и кнопки . Кнопка  служит для удаления значений.

Настройки

Настройки

Формат даты: вчера 15:30

Максимальная вложенность: 7

Максимальное количество комментариев на странице: 50

Использовать bbCode: ☐

Отображать имя пользователя, добавившего комментарий: ☒

Только для зарегистрированных пользователей: ☐

- **Формат даты** – Позволяет настроить отображение даты в модуле. Возможные значения:
 - 01.05.2013,
 - 01.05.2013 14:45,
 - 1 мая 2013 г.,
 - 1 мая,
 - 1 мая 2013, понедельник,
 - вчера 15:30,
 - не отображать.
- **Максимальная вложенность** – Ограничивает вложенность дерева комментариев.
- **Количество комментариев на странице** – Количество комментариев первого уровня, показываемых на одной странице.
- **Использовать bbCode** – Подключает форматирование комментариев с помощью bbCode.
- **Отображать имя пользователя, добавившего комментарий** – Выводит имя пользователя, добавившего комментарий на сайте
- **Только для зарегистрированных пользователей** – Параметр позволяет запретить незарегистрированным пользователям добавлять комментарии.
- **Использовать защитный код (капчу)** – Для добавления комментария пользователь должен ввести защитный код.
- **Модерация сообщений** – Если отмечена, комментарии будут отображаться на сайте только после того, как администратор вручную установит активность. Если пользователь, добавивший комментарий имеет права администратора модуля, то комментарий появится на сайте сразу.
- **Ваше сообщение уже имеется в базе** – Сообщение пользователю при попытке повторного добавления сообщения. Параметр имеет разные значения для разных языковых версий сайта.
- **Спасибо! Ваш комментарий будет проверен в ближайшее время и появится на сайте.** – Сообщение пользователю при удачном добавлении комментария. Параметр имеет разные значения для разных языковых версий сайта.
- **Подписываться на новые комментарии** – Возможность при комментировании на сайте оставить e-mail, на который будут приходить уведомления о новых

комментариях ветки.

- **Тема письма для уведомлений о новых комментариях** – Можно добавлять:
 - %title – название сайта,
 - %url – адрес сайта (например, site.ru).

Параметр имеет разные значения для разных языковых версий сайта. Параметр появляется, если отмечена опция «Подписываться на новые комментарии».

Пример:

Новый комментарий на сайте %title (%url)

- **Сообщение для уведомлений о новых комментариях** – Можно добавлять:
 - %title – название сайта,
 - %url – адрес сайта (например, site.ru),
 - %message – комментарий,
 - %link – ссылка на страницу на сайте, где комментарий отображается,
 - %actlink – ссылка для отмены подписки.

Параметр имеет разные значения для разных языковых версий сайта. Параметр появляется, если отмечена опция «Подписываться на новые комментарии».

Пример:

Здравствуйте! Вы подписались на комментарии на сайте %title (%url).
 На странице появился новый комментарий %link:
%message

 Отписаться можете по ссылке %actlink.

- **E-mail, указываемый в обратном адресе пользователю** – Возможные значения:
 - e-mail, указанный в параметрах сайта;
 - другой (при выборе этого значения появляется дополнительное поле **впишите e-mail**).

Параметр появляется, если отмечена опция «Подписываться на новые комментарии».

- **Уведомлять о поступлении новых комментариев на e-mail** – Возможность уведомления администратора о поступлении новых комментариев из формы в пользовательской части сайта.
- **E-mail для уведомлений администратора** – Возможные значения:
 - e-mail, указанный в параметрах сайта;
 - другой (при выборе этого значения появляется дополнительное поле **впишите e-mail**).

Параметр появляется, если отмечена опция «Уведомлять о поступлении новых комментариев на e-mail».

- **Тема письма для уведомлений** – Можно добавлять:
 - %title – название сайта,
 - %url – адрес сайта (например, site.ru).

Параметр появляется, если отмечена опция «Уведомлять о поступлении новых комментариев на e-mail».

Пример:

%title (%url). Новый комментарий

- **Сообщение для уведомлений** – Можно добавлять:
 - %title – название сайта,
 - %url – адрес сайта (например, site.ru),

- `%message` – комментарий,
 - `%urlpage` – ссылка на страницу на сайте, где комментарий отображается.
- Параметр появляется, если отмечена опция «Уведомлять о поступлении новых комментариев на e-mail».

Пример:

Здравствуйте, администратор сайта `%title (%url)!`
На странице `%urlpage` появился новый комментарий: `<br>%message`.

- **Уведомлять о поступлении новых комментариев по SMS** – Возможность отправлять SMS администратору при поступлении комментария. Параметр можно подключить, если в [Параметрах сайта](#) настроены SMS-уведомления.
- **Номер телефона в федеральном формате** – Номер телефона для SMS-уведомлений администратора о новом комментарии. Параметр появляется, если отмечена опция «Уведомлять о поступлении новых комментариев по SMS».
- **Сообщение для уведомлений** – Текст сообщения для SMS-уведомлений администратора о новом комментарии. Не более 800 символов. Параметр появляется, если отмечена опция «Уведомлять о поступлении новых комментариев по SMS».

Шаблонные теги

Для работы с модулем «Комментарии» служат следующие [шаблонные теги](#):

show_block – выводит последние комментарии.

Атрибуты:

- **count** – количество выводимых комментариев (по умолчанию 3);
- **modules** – модули (разделенные запятой), к которым подключены комментарии;
- **element_id** – идентификаторы элементов модулей (разделенные запятой), к которым подключены комментарии;
- **sort** – сортировка комментариев: **date** – по дате (по умолчанию), **rand** – в случайном порядке;
- **template** – шаблон тега (файл `modules/comments/views/comments.view.show_block_`**template**`.php`; по умолчанию шаблон `modules/comments/views/comments.view.show_block.php`).

Пример:

```
<insert name="show_block" module="comments" modules="shop" count="2">
```

выведет 2 последних комментария к товарам

Файлы

1. *modules/comments/admin/comments.admin.php* – редактирование комментариев;
2. *modules/comments/admin/comments.admin.config.php* – настройки модуля;
3. *modules/comments/admin/comments.admin.count.php* – количество непроверенных комментариев, если подключено модерирование комментариев, для меню административной панели;
4. *modules/comments/admin/comments.admin.inc.php* – подключение модуля к административной части других модулей;
5. *modules/comments/admin/comments.admin.param.php* – конструктор комментариев;
6. *modules/comments/admin/comments.admin.param.js* – конструктор комментариев, JS-сценарий;
7. *modules/comments/comments.php* – контроллер;
8. *modules/comments/comments.ajax.php* – обработка запроса при добавления комментария;
9. *modules/comments/comments.inc.php* – подключение модуля;
10. *modules/comments/comments.install.php* – установка модуля;
11. *modules/comments/comments.js* – JS-сценарий модуля;
12. *modules/comments/comments.model.php* – модель;
13. *modules/comments/comments.rss.php* – RSS лента комментариев;
14. *modules/comments/views/comments.view.form.php* – шаблон формы добавления комментария;
15. *modules/comments/views/comments.view.get.php* – шаблон вывода комментариев;
16. *modules/comments/views/comments.view.id.php* – шаблон одного комментария;
17. *modules/comments/views/comments.view.list.php* – шаблон списка комментариев;
18. *modules/comments/views/comments.view.show_block.php* – шаблон блока комментариев.

Рейтинг

- [Подключение](#)
- [Как добавлять рейтинг к модулю](#)
- [Административная часть](#)
- [Настройки](#)
- [Файлы](#)

Модуль относится к части CMS «Интерактив» и может быть установлен опционально.

Модуль «Рейтинг» подключается к другим модулям. В стандартной комплектации diafan.CMS «Рейтинг» подключен к следующим модулям: Фотогалерея, Новости, Статьи, Магазин, Файловый архив, Вопрос-Ответ, Объявления. В рамках модуля рейтинг подключается в настройках конкретного модуля.

В пользовательской части сайта голосование проходит с помощью технологии Ajax, то есть без перезагрузки всей страницы.

Подключение

Подключаемая часть – файл *modules/rating/rating.inc.php*. В нем описан класс *Rating_inc*. В модуле к объекту класса можно обратиться через переменную `$this->diafan->_rating`. Экземпляр класса создается при первом вызове переменной.

Методы

`string get([integer $element_id = 0], [string $module_name = ''], [integer $site_id = 0], [boolean $is_category = false])` – показывает рейтинг для элемента.

- access: public
- integer \$element_id: номер элемента модуля
- string \$module_name: название модуля
- integer \$site_id: страница сайта, к которой прикреплен элемент
- boolean \$is_category: это категория

Пример:

```
// получаем рейтинг товара в файле modules/shop/shop.model.php
$rating = $this->diafan->_rating->get($id, 'shop');
// выводим рейтинг в шаблоне модуля в файле modules/shop/views/shop.view.id.php
echo $rating;
```

Как добавлять рейтинг к модулю

В настройках модуля (*modules/модуль/admin/модуль.admin.config.php*) и в файле редактирования элементов модуля (*modules/модуль/admin/модуль.admin.php*) необходимо добавить параметр **'rating'**:

Пример:

```
public $variables = array(
    'main' => array(
        'rating' => 'module',
        ...
    ),
    ...
);
```

Административная часть

Рейтинг

☐	11.04.2013 11:57	Редактировать	Файловый архивКомпром
☐	11.04.2013 11:57	Редактировать	Фотогалерея Лошадь
☐	11.04.2013 11:57	Редактировать	Новости Двоеухи
☐	11.04.2013 11:57	Редактировать	Новости Всех с 10
☐	11.04.2013 11:57	Редактировать	Новости Новый з

☐ Отметить все На странице:

В административной части модуля собираются средние оценки за все элементы разных модулей в одну таблицу и отображаются в порядке обновления. Рейтинг отдельного элемента модуля можно посмотреть не посещая раздел «Рейтинг» в административной части, так как средняя оценка и число голосовавших за элемент отображаются при редактировании элемента модуля, к которому подключен рейтинг.

Редактировать

Объект <http://test.diafan.ru/andrey/admin/files/cat0/e>

Средняя оценка ?

Количество голосовавших ?

Дата последнего голосования ?

Кнопка ✕ служит для удаления оценки. При удалении какого-либо элемента, удаляется оценка этого элемента.

Рейтинг имеет следующие характеристики.

Основные:

- **Объект.**
- **Средняя оценка** – Числовое значение, вычисляется автоматически, как отношение суммы баллов к числу проголосовавших.
- **Количество голосовавших** – Числовое значение.
- **Дата последнего голосования** – Устанавливается после изменения рейтинга, в формате дд.мм.гггг чч:мм.

В списке оценок выводятся название и модуль оцениваемого элемента.

Настройки

Настройки

Только для зарегистрированных пользователей ☐

Защита от накруток запрещать голосовать повторно ▼

Сохранить

- **Только для зарегистрированных пользователей** – Параметр позволяет запретить неавторизованным пользователям голосовать.
- **Защита от накруток** – Возможные значения:
 - нет,
 - вести лог голосовавших,
 - запрещать голосовать повторно.

Файлы

1. *modules/rating/admin/rating.admin.php* – редактирование рейтингов;
2. *modules/rating/admin/rating.admin.config.php* – настройки модуля;
3. *modules/rating/admin/rating.admin.inc.php* – подключение модуля к административной части других модулей;

4. *modules/rating/img/** – изображения;
5. *modules/rating/rating.ajax.php* – обработка запроса при оценке элемента;
6. *modules/rating/rating.inc.php* – подключение модуля;
7. *modules/rating/rating.install.php* – установка модуля;
8. *modules/rating/rating.js* – JS-сценарий модуля;
9. *modules/rating/views/rating.view.get.php* – шаблон рейтинга элемента.

Перелинковка

- [Подключение](#)
- [Как добавлять перелинковку к модулю](#)
- [Ключевые слова](#)
- [Импорт/экспорт](#)
- [Настройки](#)
- [Файлы](#)

Модуль позволяет задать ссылку, которая будет автоматически подставляться в тексты на сайте с заданным ключевым словом или фразой. Модуль относится к части CMS «Интерактив» и может быть установлен опционально.

Модуль «Перелинковка» подключается к другим модулям. В стандартной комплектации diafan.CMS «Перелинковка» интегрирована со следующими модулями: Фотогалерея, Новости, Статьи, Вопрос-Ответ, Магазин, Файлы, Страницы сайта, Объявления. В рамках модуля перелинковка подключается в настройках конкретного модуля.

Подключение

Подключаемая часть – файл *modules/keywords/keywords.inc.php*. В нем описан класс *Keywords_inc*. В модуле к объекту класса можно обратиться через переменную `$this->diafan->_keywords`.

Экземпляр класса создается при первом вызове переменной.

Методы

string **get(string \$text)** – подставляет ключевые слова.

- access: public
- integer string \$text: исходный текст

Пример:

```
// заменяем в тексте ключевые слова на ссылки
$text = $this->diafan->_keywords->get($text);`
```

Если в настройках модуля подключен модуль «Перелинковка», то ссылки подставляются в тексты, обрабатываемые функцией [htmleditor](#).

Как добавлять перелинковку к модулю

В настройках модуля (*modules/модуль/admin/модуль.admin.config.php*) необходимо добавить параметр **'keywords'**:

Пример:

```
public $variables = array(  
    'main' => array(  
        'keywords' => 'module',  
        ...  
    ),  
    ...  
);
```

Ключевые слова

Перелинковка

Ключевые слова

Импорт/экспорт

 [Добавить ссылку](#)

☐ 2	кенгуру	/shop/animals/t
☐ 1	бумажные животные	/

☐ [Отметить все](#)

Сделать активным

Применить

 На странице: 30

Изменить

Ключевые слова и ссылки к ним можно редактировать непосредственно из списка.

Кнопка ✕ служит для удаления ключевого слова. Кнопка ⏻ служит для быстрой публикации (или, наоборот, временной деактивации) ключевого слова.

Редактировать

Ключевое слово ?

URL ?

Показывать на сайте ☒

Сохранить

Сохранить и выйти

Сохранить и добавить ещё

В

Ключевые слова имеют следующие характеристики.

Основные:

- **Ключевое слово** – Модуль найдет все слова на Вашем сайте и превратит их в ссылки на страницу, адрес которой нужно указать ниже.
- **URL** – URL-адрес страницы, куда будет вести ссылка с ключевого слова.
- **Опубликовать на сайте** – Публикация на сайте, активность. Параметр имеет разные значения для разных языковых версий сайта.

Импорт/экспорт

Импорт/экспорт

Ключевые слова Импорт/экспорт

[Экспорт](#)

Импорт:

☐ файл keywords.php (из предыдущих версий)

☐ удалить неописанные в файле строки

[Обзор...](#) [Загрузить](#)

Если в предыдущих версиях diafan.CMS вы использовали файл keywords.php, то его можно легко импортировать в новый модуль. Для этого нужно отметить опцию «файл keywords.php (из предыдущих версий)» при импорте.

Если опция не отмечена, то файл импорта должен быть следующего формата:

Пример:

```
ключевое слово 1
ссылка 1
ключевое слово 2
ссылка 2
ключевое слово 3
ссылка 3
...
```

Вы также можете скачать файл с ключевыми словами на сайте в указанном выше формате по ссылке «Экспорт».

Настройки

Настройки

Максимальное количество ссылок на странице	10
--	---------------------------------

Сохранить

В настройках можно указать:

- **Максимальное количество ссылок на странице** – Ограничивает количество ссылок на одной странице.

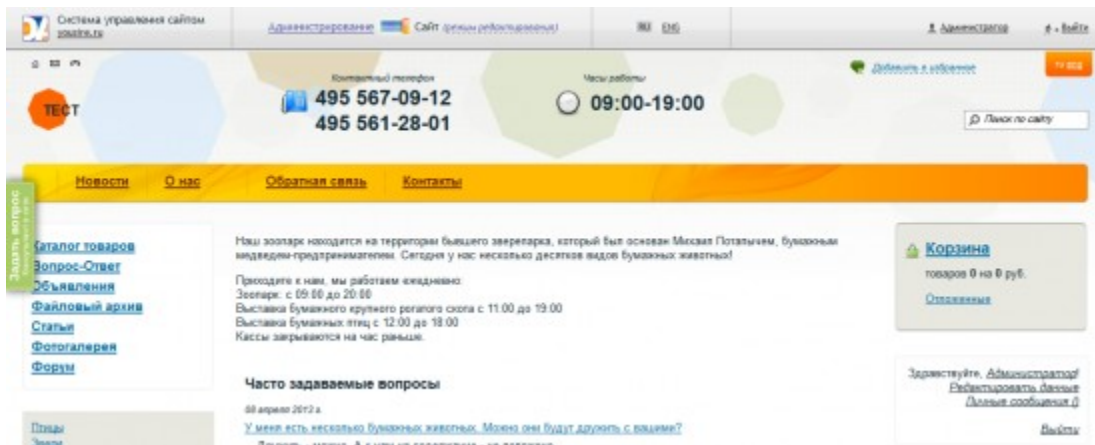
Файлы

1. *modules/keywords/admin/keywords.admin.php* – редактирование ключевых слов;
2. *modules/keywords/admin/keywords.admin.config.php* – настройки модуля;
3. *modules/keywords/admin/keywords.admin.importexport.php* – импорт и экспорт ключевых слов;
4. *modules/keywords/admin/keywords.admin.inc.php* – подключение модуля к административной части других модулей;
5. *modules/keywords/keywords.export.php* – экспорт ключевых слов;
6. *modules/keywords/keywords.inc.php* – подключение модуля;
7. *modules/keywords/keywords.install.php* – установка модуля.

Панель быстрого редактирования

- [Подключение](#)
- [Как работает панель быстрого редактирования](#)
- [Файлы](#)

Модуль позволяет редактировать данные модулей и интерфейс из пользовательской части сайта. Модуль состоит из двух частей: подключаемая часть и обработка запроса на редактирование в административной части.



Подключение

Подключаемая часть – файл `modules/useradmin/useradmin.inc.php`. В нем описан класс `Useradmin_inc`. В модуле к объекту класса можно обратиться через переменную `$this->diafan->_useradmin`. Экземпляр класса создается при первом вызове переменной.

Методы

`string get (string $text, string $name, integer $element_id, string $module_name, [integer $lang_id = 0], [string $type = ''])` – генерирует ссылку на форму редактирования.

- access: public
- string \$text: значение переменной
- string \$name: название переменной
- integer \$element_id: номер элемента
- string \$module_name: модуль
- integer \$lang_id: номер языка
- string \$type: тип данных

Пример:

```
// для администратора сделаем доступным редактирование текста новости
$text = $this->diafan->_useradmin->get($text, 'text', $id, 'news', _LANG);
```

`string get_lang (string $value, string $name, string $module_name)` – генерирует ссылку на

форму редактирования перевода.

- access: public
- string \$value: текущий перевод
- string \$name: строка для перевода
- string \$module_name: модуль

Пример:

```
$text = $this->diafan->_useradmin->get_lang('hello', 'привет', 'site');
```

string **get_meta** (string \$name, integer \$element_id, string \$module_name) – генерирует ссылку на форму редактирования meta-данных.

- access: public
- string \$name: название переменной
- integer \$element_id: номер элемента
- string \$module_name: модуль

Пример:

```
$link = $this->diafan->_useradmin->get_meta('test', $id, 'site');  
echo '<meta name="test" content="'.($link ? ' useradmin="'. $link : '').'>';
```

boolean **edit ()** – генерирует данные для формы редактирования.

- access: public

string **type (string \$name)** – возвращает тип данных (date,text,editor) по имени переменной.

- access: public
- string \$name: имя редактируемой переменной

Для разработчика

Используется в основном в [моделях](#) модулей при генерации данных, реже в [шаблоне](#) для предоставления доступа к панели быстрого редактирования.

Работа с модулем должна производиться вне зоны кэширования.

Пример:

```
// news.model.php  
$this->result = $this->diafan->_cache->get($cache_meta,  
$this->diafan->module);  
...  
foreach ($this->result["rows"] as $i => $row)  
{  
    ...  
}
```

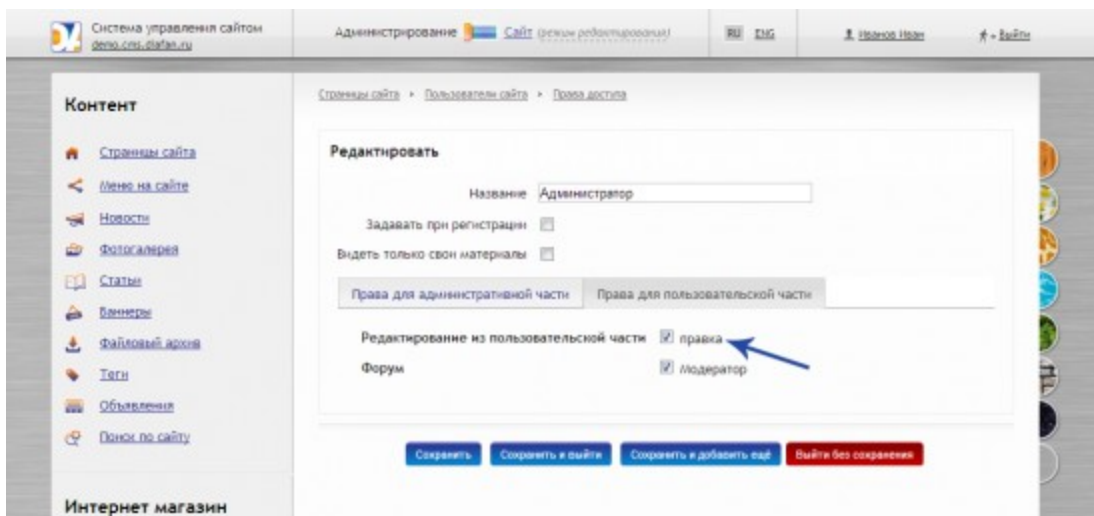
```

$this->result["rows"][$i]["name"] =
$this->diafan->_useradmin->get($row["name"], 'name', $row["id"], 'news', _LANG);
$this->result["rows"][$i]["anons"] =
$this->diafan->_useradmin->get($row["anons"], 'anons', $row["id"], 'news', _LANG);
$this->result["rows"][$i]["date"] =
$this->diafan->_useradmin->get($row["date"], 'created', $row["id"], 'news');
...
}

```

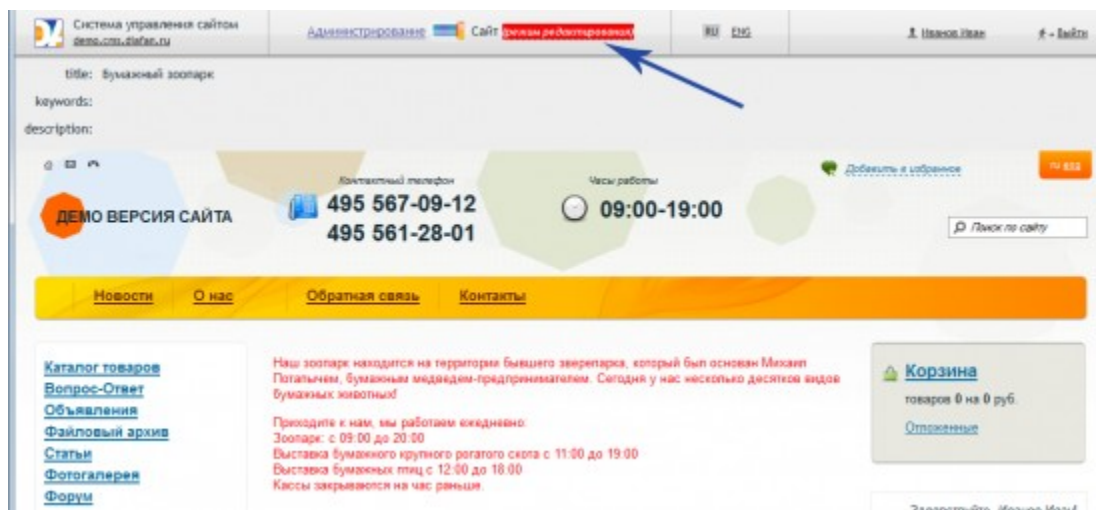
Как работает панель быстрого редактирования

Для редактирования информации из пользовательской части сайта необходимо включить эту возможность для нужного пользователя на странице редактирования пользователя в административной части. Настройка «Подключить панель быстрого редактирования» появляется только для типов пользователей, которым разрешен доступ в административную часть.

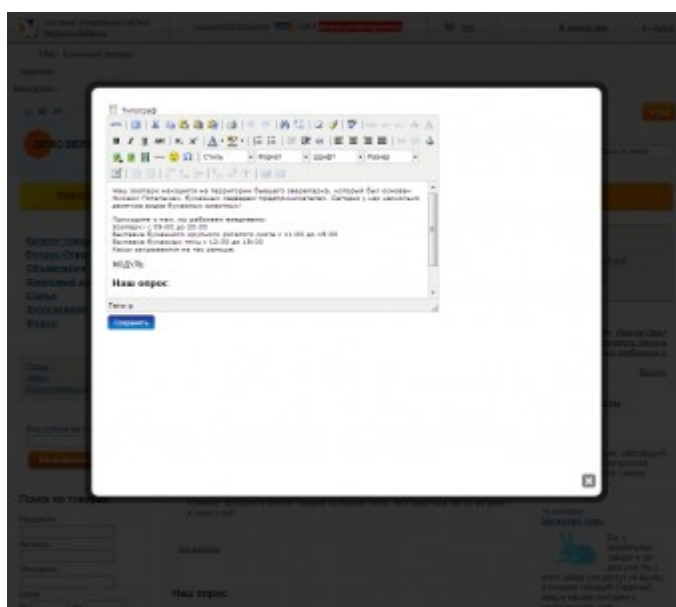


Внимание! Веб-мастер должен установить скрипты для работы панели редактирования см. [Подключение](#).

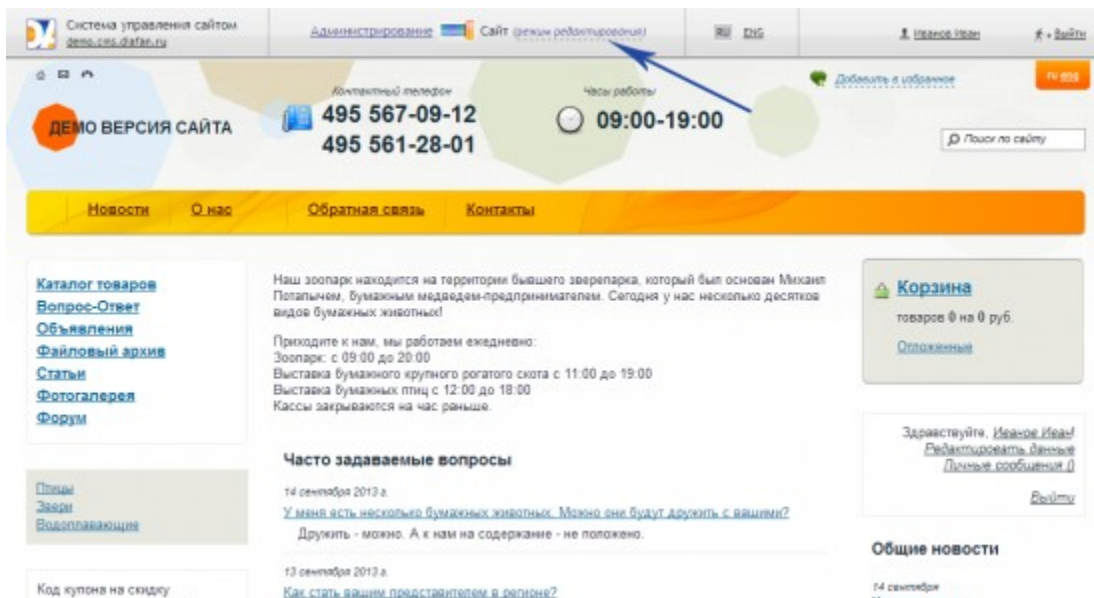
После включения режима редактирования в административной части, на сайте сверху появится административная панель. На этой панели нужно включить режим редактирования. Режим включен, когда подсвечен красным цветом.



При наведении на редактируемый элемент появится иконка «Карандаш». При нажатии на выбранный элемент откроется всплывающее окно, в котором вы сможете отредактировать элемент.



После сохранения элемента для дальнейшей навигации по сайту необходимо выключить режим редактирования. При выключенном режиме красная подсветка исчезнет.



Иконка карандаша рядом со ссылкой «режим редактирования» ведет на полное редактирование текущей страницы (категории или элемента модуля) в административной части.

Файлы

1. *modules/useradmin/admin/useradmin.admin.ajax.php* – обработка запроса на редактирование данных из пользовательской части;
2. *modules/useradmin/useradmin.ajax.php* – подгрузка панели быстрого редактирования;
3. *modules/useradmin/useradmin.css* – панель быстрого редактирования, CSS;
4. *modules/useradmin/useradmin.edit.php* – форма редактирования контента из пользовательской части;
5. *modules/useradmin/useradmin.edit.css* – форма редактирования контента из пользовательской части, CSS;
6. *modules/useradmin/useradmin.edit.js* – форма редактирования контента из пользовательской части, JS-сценарий;
7. *modules/useradmin/useradmin.inc.php* – подключение модуля;
8. *modules/useradmin/useradmin.js* – панель быстрого редактирования, JS-сценарий;
9. *modules/useradmin/views/useradmin.view.edit.php* – шаблон формы редактирования данных;
10. *modules/useradmin/views/useradmin.view.get.php* – шаблон данных, доступных для редактирования с помощью панели быстрого редактирования.

Изображения

- [Подключение](#)
- [Как добавить прикрепленные изображения к модулю](#)
- [Размеры изображений](#)
- [Настройки](#)
- [Файлы](#)

Модуль работает с изображениями, прикрепленными к элементам других модулей. Модуль состоит из двух частей: подключаемая часть и работа модуля в административной части.

Подключение

Подключаемая часть – файл *modules/images/images.inc.php*. В нем описан класс *Images_inc*. В модуле к объекту класса можно обратиться через переменную `$this->diafan->_images`. Экземпляр класса создается при первом вызове переменной.

Методы

`array get (string $variation, integer $element_id, string $module_name, integer $site_id, string $alt, [boolean $is_category = false], [integer $count = 0], [string $link_to = ""])` – получает изображения, прикрепленные к элементу модуля.

- access: public
- string \$variation: размер изображения, указанный в настройках модуля
- integer \$element_id: номер элемента, к которому прикреплены изображения
- string \$module_name: название модуля, по умолчанию модуль, прикрепленный к текущей странице
- integer \$site_id: страница сайта, к которой прикреплен элемент
- string \$alt: альтернативный текст получаемых изображений
- boolean \$is_category: изображения прикреплены к категории
- integer \$count: количество изображений
- string \$link_to_variation: размер изображения, на который ведет ссылка

`void delete(integer $element_id, string $module_name, [integer|boolean false $image_id = false])` –удаляет прикрепленные изображения.

- integer \$element_id: номер элемента, к которому прикреплены изображения
- string \$module_name: название модуля
- integer|boolean false \$image_id: номер изображения

Пример:

```
// получаем два изображения, прикрепленные к товару, ссылка с изображения ведет на
увеличенную копию изображения, в модуле «Магазин» в файле
modules/shop/shop.model.php
```



```

$images = $this->diafan->_images->get("medium", $id, "shop", $site_id, "Название товара",
false, 2, "large");

// выводим полученные файлы в шаблоне модуля в файле
modules/shop/views/shop.view.id.php
foreach($images as $img)
{
    switch($img["type"])
    {
        case 'animation':
            echo '<a href="'.BASE_PATH.$img["link"]
            .'" rel="prettyPhoto[gallery]'.$id.'.shop">';
            break;
        case 'big_image':
            echo '<a href="'.BASE_PATH.$img["link"]
            .'" rel="big_image" width="'. $img["link_width"]
            .'" height="'. $img["link_height"]."'>';
            break;
        default:
            echo '<a href="'.BASE_PATH_HREF.$img["link"]."'>';
            break;
    }
    echo ''
    . '</a> ';
}

```

Как добавить прикрепленные изображения к модулю

В настройках модуля (*modules/модуль/admin/модуль.admin.config.php*) и в файле редактирования элементов модуля (*modules/модуль/admin/модуль.admin.php*) необходимо добавить параметр **'images'**:

Пример:

```

public $variables = array(
    'main' => array(
        'images' => 'module',
        ...
    ),
    ...
);

```

Размеры изображений

Обработка изображений

 [Добавить вариант](#)

Большое изображение (полная версия)
Среднее изображение
Маленькое изображение (превью)

☐ [Отметить все](#) [Удалить](#) [Применить](#) На странице: [Изменить](#)

В административной части модуля определяются варианты изображений, которые затем используются в настройках модуля и настройках визуального редактора.

Редактировать

Название

Папка ?

Качество

Метод генерации

Размер x ?

☐ Уменьшение по большей стороне



Варианты изображений имеют следующие свойства:

Основные:

- **Название** – Название метода обработки изображений. Используется во всех модулях, где подключены изображения.
- **Папка** – Название папки латинскими буквами без пробелов, куда будут загружаться обработанные изображения. Для каждого модуля папка формируется отдельно.

Пример:

Создаем размер «Большой», папка *big*. В настройках модуля «Магазин» задаем этот вариант для полного изображения (тег *large*). Тогда увеличенное изображение

товара будет сохраняться в папке *userfiles/shop/big*.

- **Качество** – Качество сжатия файлов в формате JPEG (0 – минимальное, 100 – максимальное, 60-90 – рекомендуемое).
- **Метод обработки** – Набор действий, осуществляемых с изображением для формирования варианта изображения. Можно задать несколько действий. Действия выполняются в заданной последовательности.

Можно определить следующие действия:

- изменить пропорционально – пропорциональное уменьшение изображения. Если не отмечена опция «Уменьшение по меньшей стороне», то за основу берется минимальная пропорция, иначе максимальная;

Пример:

Необходимо, чтобы все изображения на сайте были одинакового размера: 300 на 100 пикселей. Нужно сначала пропорционально уменьшить изображение, затем обрезать по размеру 300x100. В этом случае следует отметить опцию «Уменьшение по меньшей стороне», чтобы уменьшенное изображение было больше пропорции 300x100.

Необходимо, чтобы все изображения были одной высоты – 100 пикселей. Нужно применить действие «изменить пропорционально», не отмечая опцию «Уменьшение по меньшей стороне», размеры: 1000x100. Тогда минимальной пропорцией всегда будет пропорция по высоте.

- выделить область – ручное выделение области обрезания, можно задать пропорции, при пакетной загрузке (например, при импорте товаров или при нажатии кнопки «Применить настройки ко всем ранее загруженным изображениям» действие не применяется;
- обрезать – вырезает область изображения согласно указанным размерам и отступам;
- обесцветить – переводит изображение в черно-белую цветовую гамму;
- наложить водяной знак – накладывает изображение согласно указанным расположению и смещению.

Настройки

Настройки изображений задаются в настройках модуля, к которому прикрепляются изображения. Необходимо определить следующие параметры работы модуля:

Использовать изображения – позволяет включить/отключить загрузку изображений.

Генерировать размеры изображений – размеры изображений, заданные в модуле «Изображения» и тег латинскими буквами для подключения изображения на сайте. Обязательно должны быть заданы два размера: превью изображения в списке (тег medium) и полное изображение (тег large). Параметр выводится, если отмечена опция «Использовать изображения».

Отображение изображений в списке – параметр принимает значения:

- нет (отключает отображение изображений в списке);
- показывать одно изображение;
- показывать все изображения.

Использовать изображения для категорий – позволяет включить/отключить загрузку изображений к категориям.

Генерировать размеры изображений для категорий – размеры изображений, заданные в модуле «Изображения» и тег латинскими буквами для подключения изображения на сайте. Обязательно должны быть заданы два размера: превью изображения в списке категорий (тег `medium`) и полное изображение (тег `large`). Параметр выводится, если отмечена опция «Использовать изображения для категорий».

Отображение изображений в списке категорий – параметр принимает значения:

- нет (отключает отображение изображений в списке);
- показывать одно изображение;
- показывать все изображения.

Использовать анимацию при увеличении изображений – параметр добавляет JavaScript код, позволяющий включить анимацию при увеличении изображений. Параметр выводится, если отмечена опция «Использовать изображения».

Максимальный размер загружаемых файлов – параметр показывает максимально допустимый размер загружаемых файлов, установленный в настройках хостинга. Параметр выводится, если отмечена опция «Использовать изображения».

Применить настройки ко всем ранее загруженным изображениям – позволяет перекодировать размер уже загруженных изображений. Кнопка необходима, если изменены настройки размеров изображений. Параметр выводится, если отмечена опция «Использовать изображения».

Файлы

1. *modules/images/admin/images.admin.php* – редактирование вариантов генерирования изображений;
2. *modules/images/admin/images.admin.ajax.php* – обработка Ajax-запросов при работе с изображениями в административной части;
3. *modules/images/admin/images.admin.inc.php* – подключение модуля к административной части других модулей;
4. *modules/images/admin/images.admin.inc.config.js* – подключение модуля к настройкам других модулей, JS-сценарий;
5. *modules/images/admin/images.admin.inc.js* – подключение модуля к административной части других модулей, JS-сценарий;
6. *modules/images/admin/images.admin.js* – редактирование вариантов генерирования изображений, JS-сценарий;

7. *modules/images/admin/images.admin.view.php* – шаблон вывода изображений в административной части;

8. *modules/images/images.editor.php* – плагин для визуального редактора;

9. *modules/images/images.editor.css* – плагин для визуального редактора, CSS;

10. *modules/images/images.editor.js* – плагин для визуального редактора, JS-сценарий;

11. *modules/images/images.inc.php* – подключение для работы с прикрепленными изображениями.

Прикрепленные файлы

- [Подключение](#)
- [Как добавить прикрепленные файлы к модулю](#)
- [Настройки](#)
- [Файлы](#)

Модуль работает с файлами, прикрепленными к элементам других модулей. Модуль состоит из трех частей: подключаемая часть, работа модуля в административной части и файл, отдающий прикрепленные файлы.

Подключение

Подключаемая часть – файл *modules/attachments/attachments.inc.php*. В нем описан класс *Attachments_inc*. В модуле к объекту класса можно обратиться через переменную `$this->diafan->_attachments`. Экземпляр класса создается при первом вызове переменной.

Методы

array get(integer \$element_id, string \$module_name, [integer|boolean \$attachment_id = FALSE], [integer \$param_id = 0]) – отдает информацию о прикрепленных файлах/файле.

- access: public
- integer \$element_id: номер элемента, к которому прикрепляется файл
- string \$module_name: название модуля
- integer|boolean \$attachment_id: номер файла
- integer \$param_id: номер параметра, к которому прикреплен файл

string save(integer \$element_id, string \$module_name, [array \$config = array()]) – сохраняет добавленные файлы.

- access: public
- integer \$element_id: номер элемента, к которому прикрепляется файл
- string \$module_name: название модуля
- array \$config: конфигурация

boolean|string upload(array \$file, string \$module_name, integer \$element_id, [integer|boolean \$n = FALSE], [array \$config = array()]) – загружает файлы.

- access: public
- array \$file: загружаемый файл/файлы
- string \$module_name: название модуля
- integer \$element_id: номер элемента, к которому прикрепляется файл
- integer|boolean \$n: false \$n номер файла в массиве файлов, если предан массив
- array \$config: конфигурация

boolean delete(integer \$element_id, string \$module_name, [integer|boolean \$attachment_id = FALSE], [integer \$param_id = 0]) – удаляет прикрепленные файлы/файл.

- access: public
- integer \$element_id: номер элемента, к которому прикрепляется файл
- string \$module_name: название модуля
- integer|boolean \$attachment_id: номер файла
- integer \$param_id: номер параметра, к которому прикреплен файл

Пример:

```
// получаем файлы, прикрепленные к вопросу в модуле «Вопрос-Ответ» в файле
modules/faq/faq.model.php
$attachments = $this->diafan->_attachments->get($id, 'faq');
// выводим полученные файлы в шаблоне модуля в файле
modules/faq/views/faq.view.id.php
foreach($attachments as $att)
{
    if($att["is_image"])
    {
        if($row["use_animation"])
        {
            $a_href = '<a href="'. $att["link"].'" rel="prettyPhoto[gallery'
            . $att["element_id"]. $att["module_name"]. '_1">';
            $a_href2 = '<a href="'. $att["link"].'" rel="prettyPhoto[gallery'
            . $att["element_id"]. $att["module_name"]. '_2">';
        }
        else
        {
            $a_href .= '<a href="'. $att["link"].'" rel="big_image" width="'
            . $att["width"].'" height="'. $att["height"].'">';
            $a_href2 = $a_href;
        }
        echo '<p id="attachment'. $att["id"].'">'. $a_href. $att["name"]. '</a>'
        ('. $att["size"]. ')'.
        .' '. $a_href2. '</a></p>';
    }
    else
    {
        echo '<p id="attachment'. $att["id"].'"><a href="'. $att["link"].'">'
        . $att["name"]. '</a> ('. $att["size"]. ')</p>';
    }
}
// загружаем файлы, прикрепленные к задаваемому вопросу в файле
modules/faq/faq.ajax.php
$config = array('site_id' => $site_id, 'type' => 'configmodules');
$result = $this->diafan->_attachments->save($id, "faq", $config);
```

Как добавить прикрепленные файлы к модулю

В настройках модуля (*modules/модуль/admin/модуль.admin.config.php*) и в файле редактирования элементов модуля (*modules/модуль/admin/модуль.admin.php*) необходимо добавить параметр **'attachments'**:

Пример:

```
public $variables = array(
    'main' => array(
        'attachments' => 'module',
        ...
    ),
    ...
);
```

Настройки

Настройки модуля задаются в настройках модуля, к которому прикрепляются файлы. Все настройки являются необязательными. Можно определить следующие параметры работы модуля:

Разрешить добавление файлов – позволяет пользователям прикреплять файлы к элементу модуля.

Максимальное количество добавляемых файлов – количество добавляемых файлов.

Доступные типы файлов (через запятую)

Пример:

doc, gif, jpg, mpg, pdf, png, txt, zip

Распознавать изображения – позволяет прикрепленные к сообщению файлы в формате JPEG, GIF, PNG отображать как изображения.

Размер для большого изображения – размер изображения, отображаемый в пользовательской части сайта при увеличении изображения предпросмотра.

Размер для маленького изображения – размер изображения предпросмотра.

Использовать анимацию при увеличении изображений – добавляет JavaScript-код, позволяющий включить анимацию при увеличении изображений.

Максимальный размер загружаемых файлов – параметр показывает максимально допустимый размер загружаемых файлов, установленный в настройках хостинга.

Загруженные файлы открываются по адресу:

http://www.site.ru/attachments/get/номер_файла/название_файла

Файлы

1. *modules/attachments/admin/attachments.admin.inc.php* – подключение модуля к административной части других модулей;
2. *modules/attachments/admin/attachments.admin.inc.js* – подключение модуля к

административной части других модулей, JS-сценарий;

3. *modules/attachments/attachments.get.php* – вывод прикрепленных файлов;

4. *modules/attachments/attachments.inc.php* – подключение для работы с прикрепленными файлами.

Постраничная навигация

- [Подключение](#)
- [Файлы](#)

Модуль предназначен для формирования постраничной навигации контента. Модуль состоит только из подключаемой части и шаблона.

Подключение

Подключаемая часть – файл *modules/paginator/paginator.inc.php*. В нем описан класс *Paginator_inc*. В модуле к объекту класса можно обратиться через переменную `$this->diaphan->_paginator`. Экземпляр класса создается при первом вызове переменной.

Свойства

string **\$get_nav**

- var: GET переменные, участвующие в навигации для постраничного вывода
- access: public

integer **\$nastr = 10**

- var: количество элементов, показанных на странице
- access: public

string **\$navlink**

- var: ссылка на первую страницу
- access: public

string **\$navlink_tpl**

- var: шаблон ссылки второй и последующих страниц, если не задан – используется `navlink + urlpage`
- access: public

integer **\$navnastr = 10**

- var: количество ссылок постраничной навигации, показанных на одной странице
- access: public

integer **\$nen**

- var: количество элементов в списке
- access: public

integer **\$page**

- var: номер страницы
- access: public

integer **\$polog = 0**

- var: порядковый номер элемента, с которого начинается вывод элементов
- access: public

string **\$urlpage = 'page%d/'**

- var: шаблон части ссылки, отвечающей за передачу номера страницы
- access: public

Методы

array **get()** – формирует строку навигации

- access: public

Пример:

```
// 1. Определяем свойства класса:
// номер текущей страницы
$this->diafan->_paginator->page = $this->diafan->page;

// адрес первой страницы
// вычисляем с помощью [подключения «Маршрутизатор»](/dokument/full-
manual/includes/route/)
$this->diafan->_paginator->navlink =
$this->diafan->_route->current_link("page");
// или задаем вручную
$this->diafan->_paginator->navlink = "news.html";

// шаблон последующих страниц
// вычисляем с помощью [подключения «Маршрутизатор»](/dokument/full-
manual/includes/route/)
$this->diafan->_paginator->navlink_tpl =
$this->diafan->_route->current_link("", array("page" => "%d"));
// или задаем вручную
$this->diafan->_paginator->navlink_tpl = 'news/page%d/';

// вместо переменной &nbsnavlink_tpl; можно задать utlpage, тогда navlink_tpl будет равна
navlink + urlpage
$this->diafan->_paginator->urlpage = 'page%d/';

// количество новостей
$this->diafan->_paginator->nen = DB::query_result("SELECT COUNT(*) FROM {news}");

// 2. Получаем список ссылок в постраничной навигации
$links = $this->diafan->_paginator->get();
```

```
// 3. Отправляем запрос к базе данных для получения списка новостей, ограниченных
данной страницей
$result = DB::query_range("SELECT * FROM {news}",
$this->diafan->_paginator->polog, $this->diafan->_paginator->nastr);

// 4. Подставляем полученные данные в шаблон
// в модели модуля
$this->result["paginator"] = $this->diafan->_tpl->get('get', 'paginator', $links);
// или в шаблоне модуля
echo $this->get('get', 'paginator', $links);
```

Шаблон

В модуле имеется два шаблона: *paginator.view.get.php* и *paginator.view.get_admin.php*. Первый используется для пользовательской части, второй для административной. Вы можете создать свой шаблон и использовать его в своем модуле.

Файлы

1. *modules/paginator/paginator.inc.php* – подключение для работы с постраничной навигацией;
2. *modules/paginator/views/paginator.view.get.php* – шаблон постраничной навигации для пользовательской части;
3. *modules/paginator/views/paginator.view.get_admin.php* – шаблон постраничной навигации для административной части.

Captcha

- [Подключение](#)
- [Методы](#)
- [Вопрос-Ответ](#)
- [Настройки](#)
- [Файлы](#)

САРТСНА – фильтр спам-ботов и реальных пользователей сайта, которые оставляют сообщения в формах различных модулей сайта, таких как «Обратная связь», «Комментарии», «Обратная связь», «Форум» и т.д. Капча может быть трех видов, классический «Код на картинке», более сложная вариация «reCAPTCHA», или «Вопрос-Ответ», когда администратор задает вопрос и варианты ответа, а пользователь должен выбрать правильный, чтобы его сообщение было принято на сайте.

Подключение

Подключаемая часть – файл *modules/captcha/captcha.inc.php*. В нем описан класс *Captcha_inc*. В модуле к объекту класса можно обратиться через переменную *\$this->diafan->_captcha*. Экземпляр класса создается при первом вызове переменной.

Методы

string **get**([string \$modules = "modules"], [\$error = ""], [boolean \$is_update = FALSE]) – выводит капчу.

- access: public
- string \$modules: метка капчи
- string \$error: ошибка ввода кода, если запрос передан не через Ajax
- boolean \$is_update: капча генерируется для обновления

string|boolean **is_right**([string \$modules = "modules"]) – проверяет правильность ввода капчи.

- access: public
- string \$modules: метка капчи

Вывод защитного кода в форме модуля «Комментарии».

Пример:

```
// в модели modules/comments/comments.inc.php запрашиваем капчу
$captcha = $this->diafan->_captcha->get("comments",$this->get_error("comments", "code"));

//выводим защитный код в шаблоне modules/comments/views/comments.view.form.php
echo $captcha;

//запрашиваем правильно ли введен защитный код при обработке запроса в файле
```

```
modules/faq/faq.ajax.php
$error = $this->diafan->_captcha->is_right('faq');
if($error)
{
    // защитный код введен неправильно
}
```

Вопрос-Ответ

Бэкенд «Вопрос-Ответ» имеет свой административный интерфейс для ввода вопросов. Для добавления вопроса нужно заполнить следующие поля.

Основные:

- **Вопрос** – Вопрос, на который должен ответить посетитель сайта для того, чтобы пройти проверку. Параметр имеет разные значения для разных языковых версий сайта.
- **Опубликовать на сайте** – Если не отмечено, то вопрос не будет участвовать в проверке. Параметр имеет разные значения для разных языковых версий сайта.
- **Ответы** – Возможные варианты ответов с указанием правильного.

Настройки

В настройках модуля можно выбрать бэкенд для капчи и вписать настройки для бэкенда.

- **Тип** – Выбор метода фильтрации спам-ботов. Возможные значения:
 - Код на картинке,
 - reCAPTCHA,
 - Вопрос-Ответ.
- **Public Key для сервиса [reCAPTCHA](#)** – Параметр выводится, если в поле «Тип» выбрано «reCAPTCHA».
- **Private Key для сервиса [reCAPTCHA](#)** – Параметр выводится, если в поле «Тип» выбрано «reCAPTCHA».

Файлы

1. *modules/captcha/admin/captcha.admin.php* – редактирование вопросов для капчи;
2. *modules/captcha/admin/captcha.admin.config.php* – настройки модуля;
3. *modules/captcha/admin/captcha.admin.config.js* – настройка модуля, JS-сценарий;
4. *modules/captcha/admin/captcha.admin.js* – редактирование вопросов для капчи, JS-сценарий;
5. *modules/captcha/captcha.get.php* – генерирование изображения капчи;
6. *modules/captcha/captcha.inc.php* – подключение для работы с каптчей;

7. *modules/captcha/captcha.install.php* – установка модуля;
8. *modules/captcha/views/captcha.view.get.php* – шаблон стандартной каптчи;
9. *modules/captcha/views/captcha.view.get_qa.php* – шаблон каптчи «Вопрос-Ответ»;
10. *modules/captcha/views/captcha.view.get_recaptcha.php* – шаблон reCAPTCHA.

Разработчикам

Данный раздел документации предназначен для программистов, знакомых с языком программирования PHP, разбирающихся в ООП, и, конечно, «дружащих» с HTML и JS.

- [Основные принципы diafan.CMS](#)
- [Архитектура diafan.CMS](#)
- [Административная часть сайта](#)
- [Подключения \(API\)](#)
- [Список внешних плагинов](#)

Основные принципы diafan.CMS

- diafan.CMS призвана облегчать разработку сайта, а не усложнять ее.
- При любых вносимых в скрипты изменениях [отключайте кеширование и включайте режим разработчика](#).
- Все файлы diafan.CMS сохранены в формате *UTF-8 without BOM*. **Встроенный в Windows «Блокнот» не работает с этим форматом!** Настоятельно рекомендуем для правки любых файлов использовать бесплатный редактор [Notepad++](#), который однозначно и бесспорно более удобный.
- Основа в diafan.CMS – [страницы сайта](#) и [меню](#). Это основной скелет сайта, состоящий из дерева страниц.
На их основе формируются публичные адреса страниц сайта (ЧПУ).
Свойства страниц – это заголовки title, тексты и т.д.
На страницы сайта навешиваются [модули](#).
- Модуль сам по себе в пользовательской части сайта существовать не может. Чтобы модуль заработал на сайте, он должен быть прикреплен к какой-нибудь странице сайта.

Пример:

Создается страница «Новая», имеющая свои *title*, *keywords*, *description* и адрес *site.ru/new/*, и только затем на эту страницу можно «повесить» модуль, например, «Новости». Модуль располагается на странице *site.ru/new/* и создает вглубь уже свои страницы со своими адресами, например, *site.ru/new/nowost1/* и *site.ru/new/nowost2/* и *site.ru/new/page2/* и т.д.

- Общее оформление (дизайн) сайта находится в файле (файлах) *themes/site.php* (и прочих в папке *themes*). [Файл шаблон дизайна по умолчанию – site.php](#). То есть, если нужно подправить шапку на сайте или высоту подвала, нужно идти в файл *themes/site.php*.
- Чтобы дизайн какой-то страницы на сайте был отличным от основного дизайна, нужно создать в папке *themes* еще один файл-шаблон и затем при редактировании нужной страницы выбрать в выпадающем списке «Дизайн страницы» этот файл.
- [Оформление модулей](#) находится в папке модуля в файле *modules/имя_модуля/views/имя_модуля.view.название_шаблона.php*. То есть, если нужно подправить карточку товара и, например, передвинуть кнопку «Купить», нужно идти в файл *modules/shop/views/shop.view.id.php*.

Архитектура diafan.CMS

Все запросы к системе попадают в первую очередь на файл *index.php*. Файл выполняет следующие функции:

1. подключение всех «жизненно важных» файлов;
2. проверка наличия папки инсталляции – если да, то перенаправление на установку системы;
3. проверка запроса на доступ к административной папке – если да, то перенаправление на [административную часть](#);
4. определение путей сайта (анализируется адрес, который после *site.ru/*);
5. инициация основного объекта системы.

Основной объект системы – экземпляр класса *Init*, описанный в файле *includes/init.php*.

При инициации основного объекта системы происходит:

1. старт сессии;
2. определение пользователя;
3. определение запрошенной языковой версии сайта;
4. определение запрошенной страницы сайта;
5. подключение возможного модуля(лей);
6. подключение шаблона дизайна сайта (*themes/site.php* по умолчанию).

Экземпляр класса *Init* передается во все другие объекты и доступен из любого места системы в виде переменной [\\$this->diafan](#).

Таким образом, в любом месте системы можно получить глобальные данные:

1. данные о текущей странице;
2. переданные в URL переменные;
3. общие функции;
4. объекты классов-подключений.

Теперь подробнее об этих данных:

Данные о текущей странице

В область глобальной видимости попадают данные о текущей странице и данные, сформированные модулем (это PHP-переменные, которые можно использовать как угодно в любом месте diafan.CMS):

- [\\$this->diafan->cid](#) – id-номер текущей страницы, уникальный номер каждой страницы сайта;
- [\\$this->diafan->rewrite](#) – ЧПУ текущей страницы сайта, для страницы *site.ru/news/popular/novost/* в переменной будет «news/popular/novost»;
- [\\$this->diafan->rewrite_module](#) – ЧПУ текущей страницы, сгенерированный модулем, для страницы *site.ru/news/popular/novost/* в переменной будет «popular/novost»;
- [\\$this->diafan->name](#) – название текущей страницы;
- [\\$this->diafan->title_meta](#) – заголовок текущей страницы из тега *title*;
- [\\$this->diafan->titlemodule](#) – заголовок страницы, сформированный автоматически

прикрепленным модулем;

- `$this->diafan->edit_titlemodule` – данные для редактирования заголовка модуля;
- `$this->diafan->titlemodule_meta` – заголовок текущей страницы для тега *title*, сформированный прикрепленным модулем;
- `$this->diafan->keywords` – данные из тега *keywords* текущей страницы;
- `$this->diafan->descr` – данные из тега *descriptions* текущей страницы;
- `$this->diafan->theme` – имя файла шаблона дизайна текущей страницы;
- `$this->diafan->text` – контент текущей страницы;
- `$this->diafan->hide_previous_next` – скрыть ссылки на предыдущую, последующую страницы
- `$this->diafan->parent_id` – номер страницы родителя;
- `$this->diafan->title_no_show` – не показывать заголовок *h1* текущей страницы, *да/нет* (1/0);
- `$this->diafan->timeedit` – время редактирования текущей страницы, в UNIX-формате;
- `$this->diafan->module` – название модуля, прикрепленного к текущей странице;
- `$this->diafan->path` – часть навигации «Хлебные крошки», сформированная прикрепленным модулем.

Переданные в URL переменные

Список всех доступных переменных задан в массиве `$this->diafan->rewrite_variable_names_all`.

- `$this->diafan->page` – номер страницы при пагинации, например, для *site.ru/news/page2/* в переменной будет "2";
- `$this->diafan->show` – номер элемента, если отключено автоматическое формирование ЧПУ для модулей и страница имеет вид *site.ru/shop/cat3/show3/*, в переменной будет "3";
- `$this->diafan->add` – *TRUE* при добавлении нового элемента, например, комментария;
- `$this->diafan->edit` – номер элемента для редактирования, например, при редактировании своего сообщения на форуме;
- `$this->diafan->param` – номер параметра, например, при поиске в магазине;
- `$this->diafan->comments` – номер страницы с комментариями;
- `$this->diafan->cat` – номер категории, если отключено автоматическое формирование ЧПУ для модулей и страница имеет вид *site.ru/shop/cat5/* в переменной будет "5";
- `$this->diafan->year` – выведет текущий год;
- `$this->diafan->month` – выведет текущий месяц;
- `$this->diafan->step` – шаг, переменная аналог *page*, например для корзины заказов в магазине. На странице *site.ru/order/step2/* в переменной будет "2";
- `$this->diafan->sort` – сортировка.

Общие функции

Класс *Init* наследует абстрактный класс *Core* (*includes/core.php*). Все функции этого класса можно вызвать в контроллере и модели модуля в виде:

```
$this->diafan->название_функции();
```

Объекты классов-подключений

В основном объекте иницируются классы:

- пользователи (класс `User`, файл `includes/user.php`, вызывается `$this->diafan->_user`);
- кеширование (класс `Cache`, файл `includes/cache.php`, вызывается `$this->diafan->_cache`);
- маршрутизатор (класс `Route`, файл `includes/route.php`, вызывается `$this->diafan->_route`);
- подключение шаблонов (класс `Template`, файл `includes/template.php`, вызывается `$this->diafan->_tpl`).

Кроме того, через основной объект можно обратиться к любому модулю-подключению. Если в модуле есть файл `modules/модуль/модуль.inc.php`, где описан класс `Модуль_inc`, то при первом обращении к переменной `$this->diafan->_модуль`, создастся экземпляр класса.

Структура базы данных diafan.CMS

diafan.CMS состоит из следующих таблиц базы данных.

{access}

Таблица ядра системы. Хранит информацию о правах доступа типов пользователей к отдельным элементам модулей

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
element_id	int(11)	идентификатор элемента модуля (0, если задано cat_id)
cat_id	int(11)	идентификатор категории модуля (0, если задано element_id)
module_name	varchar(25)	модуль
role_id	int(3)	идентификатор типа пользователя, которому дается доступ
trash	enum('0', '1')	запись удалена в корзину: 0 - нет, 1 - да

{adminsitem}

Таблица ядра системы. Хранит информацию о меню административной части сайта

Поле	Тип (размерность)	Описание
id	tinyint(3)	идентификатор
parent_id	int(11)	идентификатор родителя
count_children	tinyint(3)	количество вложенных записей
group_id	enum('1','2','3','4')	поле
name	varchar(100)	поле
rewrite	varchar(30)	поле
act	enum('0','1')	показывать на сайте: 0 - нет, 1 - да

sort	tinyint(3)	поле
docs	varchar(255)	ссылка на раздел в документации

{adminsitere_parents}

Таблица связей принадлежности пунктов друг другу (вложений)

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
element_id	int(11)	идентификатор элемента модуля
parent_id	int(11)	идентификатор родителя

{ads}

Таблица модуля «Объявления». Хранит объявления

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
nameНОМЕР_ЯЗЫКА	text	название
actНОМЕР_ЯЗЫКА	enum('0','1')	показывать на сайте: 0 - нет, 1 - да
date_start	int(10)	дата начала показа
date_finish	int(10)	дата окончания показа
created	int(10)	дата создания в формате UNIXTIME
map_no_show	enum('0','1')	не показывать на карте сайта: 0 - нет, 1 - да
changefreq	enum('always', 'hourly', 'daily', 'weekly', 'monthly', 'yearly', 'never')	Changefreq
priority	varchar(3)	Priority
site_id	int(11)	идентификатор страницы сайта из таблицы

[{site}](#)

keywords	НОМЕР_ЯЗЫКА	varchar(250)	ключевые слова, тэг Keywords
descr	НОМЕР_ЯЗЫКА	text	описание, тэг Description
title_meta	НОМЕР_ЯЗЫКА	varchar(250)	заголовок окна в браузере, тэг Title
anons	НОМЕР_ЯЗЫКА	text	анонс
text	НОМЕР_ЯЗЫКА	text	описание
prior		enum('0', '1')	важно, всегда сверху: 0 - нет, 1 - да
timeedit		int(12)	время последнего изменения в формате UNIXTIME
access		enum('0', '1')	доступ ограничен: 0 - нет, 1 - да
admin_id		int(11)	пользователь, добавивший или первый отредактировавший объявление в административной части, из таблицы {users}
user_id		int(11)	пользователь, добавивший объявление через форму на сайте, из таблицы {users}
theme		varchar(50)	шаблон страницы сайта
view		varchar(50)	шаблон модуля
trash		enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{ads_category}

Таблица модуля «Объявления». Хранит категории объявлений

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
name	НОМЕР_ЯЗЫКА text	название

actHOMEP_ЯЗЫКА	enum('0','1')	показывать на сайте: 0 - нет, 1 - да
map_no_show	enum('0','1')	не показывать на карте сайта: 0 - нет, 1 - да
changefreq	enum('always', 'hourly', 'daily', 'weekly', 'monthly', 'yearly', 'never')	Changefreq
priority	varchar(3)	Priority
parent_id	int(11)	идентификатор родителя
count_children	tinyint(3)	количество вложенных записей
site_id	int(11)	идентификатор страницы сайта из таблицы {site}
keywordsHOMEP_ЯЗЫКА	varchar(250)	ключевые слова, тэг Keywords
descrHOMEP_ЯЗЫКА	text	описание, тэг Description
title_metaHOMEP_ЯЗЫКА	varchar(250)	заголовок окна в браузере, тэг Title
anonsHOMEP_ЯЗЫКА	text	анонс
textHOMEP_ЯЗЫКА	text	описание
sort	int(11)	порядковый номер для сортировки
timeedit	int(12)	время последнего изменения в формате UNIXTIME
access	enum('0', '1')	доступ ограничен: 0 - нет, 1 - да
admin_id	int(11)	пользователь, добавивший или первый отредактировавший категорию в административной части, из таблицы {users}
theme	varchar(50)	шаблон страницы сайта
view	varchar(50)	шаблон модуля
view_element	varchar(50)	шаблон модуля для элементов в категории

trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да
-------	---------------	---

{ads_category_parents}

Таблица модуля «Объявления». Хранит информацию о вложенностях категорий объявлений друг в друга

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
element_id	int(11)	идентификатор категории из таблицы {ads_category}
parent_id	int(11)	идентификатор категории-родителя из таблицы {ads_category}
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{ads_category_rel}

Таблица модуля «Объявления». Хранит информацию о принадлежности объявлений к категориям

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
element_id	int(11)	идентификатор объявления из таблицы {ads}
cat_id	int(11)	идентификатор категории из таблицы {ads_category}
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{ads_counter}

Таблица модуля «Объявления». Хранит информацию о просмотрах объявлений

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
element_id	int(11)	идентификатор объявления из таблицы {ads}

count_view	smallint(11)	количество просмотров
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{ads_param}

Таблица модуля «Объявления». Хранит названия дополнительных характеристик

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
nameНОМЕР_ЯЗЫКА	varchar(250)	поле
type	varchar(20)	поле
sort	int(11)	порядковый номер характеристики для сортировки
search	enum('0','1')	характеристика выводиться в форме поиска: 0 - нет, 1 - да
list	enum('0','1')	характеристика выводиться в списке: 0 - нет, 1 - да
block	enum('0','1')	характеристика выводиться в блоке объявлений: 0 - нет, 1 - да
required	enum('0','1')	обязательно для заполнения: 0 - нет, 1 - да
page	enum('0','1')	отдельная страница для значений: 0 - нет, 1 - да
textНОМЕР_ЯЗЫКА	text	описание
config	text	
display_in_sort	enum('0','1')	выводиться в блоке для сортировки: 0 - нет, 1 - да
measure_unitНОМЕР_ЯЗЫКА	varchar(50)	
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{ads_param_category_rel}

Таблица модуля «Объявления». Хранит информацию о принадлежности характеристики к категории объявлений

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
element_id	int(11)	идентификатор характеристики из таблицы {ads_param}
cat_id	int(11)	идентификатор категории из таблицы {ads_category}
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{ads_param_element}

Таблица модуля «Объявления». Хранит значения дополнительных характеристик

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
valueНОМЕР_ЯЗЫКА	text	значение характеристики
param_id	int(11)	идентификатор характеристики из таблицы {ads_param}
element_id	int(11)	идентификатор объявления из таблицы {ads}
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{ads_param_select}

Таблица модуля «Объявления». Хранит значения дополнительных характеристик с типом «выпадающий список» и «галочка»

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
param_id	int(11)	идентификатор характеристики из таблицы {ads_param}

value	tinyint(2)	значение для типа характеристики «галочка»: 0 - нет, 1 - да
nameНОМЕР_ЯЗЫКА	varchar(50)	значение
sort	int(11)	порядковый номер для сортировки
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{ads_rel}

Таблица модуля «Объявления». Хранит информацию о «похожих» объявлениях

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
element_id	int(11)	идентификатор объявления из таблицы {ads}
rel_element_id	int(11)	идентификатор похожего объявления из таблицы {ads}
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{attachments}

Таблица модуля «Прикрепленные файлы». Хранит информацию о загруженных пользователями файлах (комментарии, вопрос-ответ, форум и т.д.)

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
name	varchar(50)	поле
module_name	varchar(10)	название модуля, к которому прикреплен файл
element_id	int(11)	идентификатор элемента модуля, к которому прикреплен файл
param_id	int(11)	идентификатор поля или характеристики с типом «файлы»
extension	varchar(255)	расширение файла
size	int(11)	размер файла

is_image	enum('0','1')	это изображение: 0 - нет, 1 - да
access_admin	enum('0','1')	доступ только для администратора: 0 - нет, 1 - да
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{banners}

Таблица модуля «Баннеры». Хранит баннеры

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
name	text	название
altНОМЕР_ЯЗЫКА	varchar(250)	атрибут alt для баннера-изображения
titleНОМЕР_ЯЗЫКА	varchar(250)	атрибут title для баннера-изображения
actНОМЕР_ЯЗЫКА	enum('0','1')	показывать на сайте: 0 - нет, 1 - да
type	enum('1','2','3')	тип: 1 - изображение, 2 - флэш, 3 - HTML
file	varchar(250)	
html	text	HTML код баннера-блока
link	text	ссылка на баннер-изображение или баннер-флэш
cat_id	int(11)	идентификатор категории баннеров из таблицы {banners_category}
check_number	enum('0','1')	ограничить количество показов: 0 - нет, 1 - да
check_click	enum('0','1')	ограничить по количеству кликов: 0 - нет, 1 - да
show_click	int(10)	ограничить по количеству кликов: осталось кликов
date_start	int(10)	дата начала показа
date_finish	int(10)	дата окончания показа

show_number	int(10)	ограничить количество показов: осталось показов
click	int(10)	всего кликов
width	int(10)	ширина флэш-баннера
height	int(10)	высота флэш-баннера
check_user	enum('0','1')	ограничить количество показов посетителю в сутки: 0 - нет, 1 - да
show_user	int(10)	количество показов посетителю в сутки
count_view	int(10)	всего показов
target_blank	enum('0','1')	открывать ссылку в новом окне
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{banners_category}

Таблица модуля «Баннеры». Хранит категории баннеров

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
name	text	название
act	enum('0','1')	показывать на сайте: 0 - нет, 1 - да
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{banners_category_rel}

Таблица модуля «Баннеры». Хранит информацию о принадлежности баннера категориям (если они подключены)

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
element_id	int(11)	идентификатор баннера из таблицы {banners}

cat_id	int(11)	идентификатор категории баннеров из таблицы {banners_category}
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{banners_site_rel}

Таблица модуля «Баннеры». Хранит информацию о том, на каких страницах сайта выводиться баннер

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
element_id	int(11)	идентификатор баннера из таблицы {banners}
site_id	int(11)	идентификатор страницы сайта из таблицы {site}
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{clauses}

Таблица модуля «Статьи». Хранит статьи

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
nameНОМЕР_ЯЗЫКА	text	название
actНОМЕР_ЯЗЫКА	enum('0','1')	показывать на сайте: 0 - нет, 1 - да
date_start	int(10)	дата начала показа
date_finish	int(10)	дата окончания показа
map_no_show	enum('0','1')	не показывать на карте сайта: 0 - нет, 1 - да
changefreq	enum('always', 'hourly', 'daily', 'weekly', 'monthly', 'yearly', 'never')	Changefreq
priority	varchar(3)	Priority
created	int(10)	дата создания

cat_id	int(11)	идентификатор категории
site_id	int(11)	идентификатор страницы сайта из таблицы {site}
keywordsHOMEP_ЯЗЫКА	varchar(250)	ключевые слова, тэг Keywords
descrHOMEP_ЯЗЫКА	text	описание, тэг Description
title_metaHOMEP_ЯЗЫКА	varchar(250)	заголовок окна в браузере, тэг Title
anonsHOMEP_ЯЗЫКА	text	анонс
textHOMEP_ЯЗЫКА	text	описание
prior	enum('0','1')	важно, всегда сверху: 0 - нет, 1 - да
timeedit	int(12)	время последнего изменения в формате UNIXTIME
access	enum('0', '1')	доступ ограничен: 0 - нет, 1 - да
admin_id	int(11)	пользователь, добавивший или первый отредактировавший статью в административной части, из таблицы {users}
theme	varchar(50)	шаблон страницы сайта
view	varchar(50)	шаблон модуля
sort	int(11)	порядковый номер для сортировки
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{clauses_category}

Таблица модуля «Статьи». Хранит категории статей (если они подключены)

Поле	Тип (размерность)	Описание
------	-------------------	----------

id	int(11)	идентификатор
nameHOMEP_ЯЗЫКА	text	название
actHOMEP_ЯЗЫКА	enum('0','1')	показывать на сайте: 0 - нет, 1 - да
map_no_show	enum('0','1')	не показывать на карте сайта: 0 - нет, 1 - да
changefreq	enum('always', 'hourly', 'daily', 'weekly', 'monthly', 'yearly', 'never')	Changefreq
priority	varchar(3)	Priority
parent_id	int(11)	идентификатор родителя
count_children	tinyint(3)	количество вложенных записей
site_id	int(11)	идентификатор страницы сайта из таблицы {site}
keywordsHOMEP_ЯЗЫКА	varchar(250)	ключевые слова, тэг Keywords
descrHOMEP_ЯЗЫКА	text	описание, тэг Description
title_metaHOMEP_ЯЗЫКА	varchar(250)	заголовок окна в браузере, тэг Title
anonsHOMEP_ЯЗЫКА	text	анонс
textHOMEP_ЯЗЫКА	text	описание
sort	int(11)	порядковый номер для сортировки
timeedit	int(12)	время последнего изменения в формате UNIXTIME
access	enum('0', '1')	доступ ограничен: 0 - нет, 1 - да
admin_id	int(11)	пользователь, добавивший или первый отредктировавший категорию в административной части, из таблицы {users}

theme	varchar(50)	шаблон страницы сайта
view	varchar(50)	шаблон модуля
view_element	varchar(50)	шаблон модуля для элементов в категории
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{clauses_category_parents}

Таблица модуля «Статьи». Хранит информацию о вложенностях категорий статей друг в друга (если они подключены)

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
element_id	int(11)	идентификатор элемента модуля
parent_id	int(11)	идентификатор родителя
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{clauses_category_rel}

Таблица модуля «Статьи». Хранит информацию о принадлежности статьи категориям (если они подключены)

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
element_id	int(11)	идентификатор элемента модуля
cat_id	int(11)	идентификатор категории
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{clauses_counter}

Таблица модуля «Статьи». Хранит информацию о просмотрах статей

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор

element_id	int(11)	идентификатор объявления из таблицы {clauses}
count_view	smallint(11)	количество просмотров
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{clauses_rel}

Таблица модуля «Статьи». Хранит информацию о «похожих» статьях

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
element_id	int(11)	идентификатор статьи из таблицы {clauses}
rel_element_id	int(11)	идентификатор похожей статьи из таблицы {clauses}
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{comments}

Таблица модуля «Комментарии». Хранит комментарии

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
user_id	int(11)	идентификатор пользователя, создавшего комментарий, из таблицы {users}
text	text	комментарий
act	enum('0','1')	показывать на сайте: 0 - нет, 1 - да
created	int(10)	дата создания
module_name	varchar(17)	название модуля, к которому прикреплен комментарий
element_id	int(11)	идентификатор элемента модуля, к которому прикреплен комментарий
parent_id	int(11)	идентификатор комментария-родителя

count_children	tinyint(3)	количество вложенных комментариев
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{comments_param}

Таблица модуля «Комментарии». Хранит поля из конструктора комментариев

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
nameНОМЕР_ЯЗЫКА	varchar(250)	название поля
type	varchar(20)	тип поля
module_name	varchar(17)	название модуля, для которого поле выводиться
sort	int(11)	порядковый номер для сортировки
required	enum('0','1')	обязательно для заполнения: 0 - нет, 1 - да
show_in_list	enum('0','1')	показывать в списке: 0 - нет, 1 - да
show_in_form_auth	enum('0','1')	показывать в форме авторизованным пользователям: 0 - нет, 1 - да
show_in_form_no_auth	enum('0','1')	показывать в списке: 0 - нет, 1 - да
textНОМЕР_ЯЗЫКА	text	описание
config	text	настройки поля
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{comments_param_element}

Таблица модуля «Комментарии». Хранит значения полей из конструктора комментариев при добавлении комментариев

Поле	Тип (размерность)	Описание
------	-------------------	----------

id	int(11)	идентификатор
value	text	значение
param_id	int(11)	номер поля из таблицы {comments_param}
element_id	int(11)	идентификатор комментария из таблицы {comments}
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{comments_param_select}

Таблица модуля «Комментарии». Хранит название для полей типа select из конструктора комментариев

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
param_id	int(11)	номер поля из таблицы {comments_param}
value	tinyint(2)	значение для типа характеристики «галочка»: 0 - нет, 1 - да
nameНОМЕР_ЯЗЫКА	varchar(50)	поле
sort	int(11)	порядковый номер для сортировки
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{comments_parents}

Таблица модуля «Комментарии». Хранит информацию о вложенностях комментариев друг в друга

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
element_id	int(11)	идентификатор комментария из таблицы {comments}
parent_id	int(11)	

идентификатор комментария-родителя из таблицы [{comments}](#)

trash enum('0','1') запись удалена в корзину: 0 - нет, 1 - да

{config}

Таблица ядра. Хранит значения настроек разных модулей сайта

Поле	Тип (размерность)	Описание
id	tinyint(3)	идентификатор
module_name	varchar(12)	название модуля, для которого определена настройка
name	varchar(50)	название настройки
lang_id	tinyint(3)	номер языка, если настройка переводиться для языковых зеркал
value	text	значение настройки
site_id	int(11)	идентификатор страницы сайта, к которой прикреплен модуль, из таблицы {site}

{faq}

Таблица модуля «Вопрос-Ответ». Хранит вопросы и ответы

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
nameНОМЕР_ЯЗЫКА	text	имя задавшего вопрос
actНОМЕР_ЯЗЫКА	enum('0','1')	показывать на сайте: 0 - нет, 1 - да
date_start	int(10)	дата начала показа
date_finish	int(10)	дата окончания показа
map_no_show	enum('0','1')	не показывать на карте сайта: 0 - нет, 1 - да
changefreq	enum('always', 'hourly',	Changefreq

	'daily', 'weekly', 'monthly', 'yearly', 'never')	
priority	varchar(3)	Priority
created	int(10)	дата создания
mail	varchar(40)	е-mail пользоваля, задавшего вопрос
cat_id	int(11)	идентификатор категории из таблицы {faq_category}
site_id	int(11)	идентификатор страницы сайта из таблицы {site}
keywordsНОМЕР_ЯЗЫКА	varchar(250)	ключевые слова, тэг Keywords
descrНОМЕР_ЯЗЫКА	text	описание, тэг Description
title_metaНОМЕР_ЯЗЫКА	varchar(250)	заголовок окна в браузере, тэг Title
anonsНОМЕР_ЯЗЫКА	text	вопрос
textНОМЕР_ЯЗЫКА	text	ответ
often	char(1)	часто задаваемый вопрос: 0 - нет, 1 - да
timeedit	int(12)	время последнего изменения в формате UNIXTIME
access	enum('0', '1')	доступ ограничен: 0 - нет, 1 - да
admin_id	int(11)	пользователь, добавивший или первый отредактировавший вопрос в административной части, из таблицы {users}
user_id	int(11)	пользователь, добавивший вопрос из формы на сайте, из таблицы {users}
theme	varchar(50)	шаблон страницы сайта
view	varchar(50)	шаблон модуля

trash

enum('0','1')

запись удалена в корзину: 0 - нет, 1 - да

{faq_category}

Таблица модуля «Вопрос-Ответ». Хранит категории вопросов (если они подключены)

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
nameНОМЕР_ЯЗЫКА	text	название
actНОМЕР_ЯЗЫКА	enum('0','1')	показывать на сайте: 0 - нет, 1 - да
map_no_show	enum('0','1')	не показывать на карте сайта: 0 - нет, 1 - да
changefreq	enum('always', 'hourly', 'daily', 'weekly', 'monthly', 'yearly', 'never')	Changefreq
priority	varchar(3)	Priority
parent_id	int(11)	идентификатор родителя из таблицы {faq_category}
count_children	tinyint(3)	количество вложенных категорий
site_id	int(11)	идентификатор страницы сайта из таблицы {site}
keywordsНОМЕР_ЯЗЫКА	varchar(250)	ключевые слова, тэг Keywords
descrНОМЕР_ЯЗЫКА	text	описание, тэг Description
title_metaНОМЕР_ЯЗЫКА	varchar(250)	заголовок окна в браузере, тэг Title
anonsНОМЕР_ЯЗЫКА	text	анонс
textНОМЕР_ЯЗЫКА	text	описание
sort	int(11)	порядковый номер для сортировки

timeedit	int(12)	время последнего изменения в формате UNIXTIME
access	enum('0', '1')	доступ ограничен: 0 - нет, 1 - да
admin_id	int(11)	пользователь, добавивший или первый отредактировавший категорию в административной части, из таблицы {users}
theme	varchar(50)	шаблон страницы сайта
view	varchar(50)	шаблон модуля
view_element	varchar(50)	шаблон модуля для элементов в категории
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{faq_category_parents}

Таблица модуля «Вопрос-ответ». Хранит информацию о вложенностях категорий вопросов друг в друга (если они подключены)

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
element_id	int(11)	идентификатор категории из таблицы {faq_category}
parent_id	int(11)	идентификатор категории-родителя из таблицы {faq_category}
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{faq_category_rel}

Таблица модуля «Вопрос-ответ». Хранит информацию о принадлежности статьи категориям (если они подключены)

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
element_id	int(11)	идентификатор вопроса из таблицы {faq}
cat_id	int(11)	идентификатор категории из таблицы {faq_category}

trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да
-------	---------------	---

{faq_counter}

Таблица модуля «Вопрос-Ответ». Хранит информацию о просмотрах вопросов

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
element_id	int(11)	идентификатор объявления из таблицы {faq}
count_view	smallint(11)	количество просмотров
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{faq_rel}

Таблица модуля «Вопрос-Ответ».Хранит информацию о «похожих» вопросах

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
element_id	int(11)	идентификатор вопроса из таблицы {faq}
rel_element_id	int(11)	идентификатор похожего вопроса из таблицы {faq}
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{feedback}

Таблица модуля «Обратная связь». Хранит сообщения

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
created	int(10)	дата создания
site_id	int(11)	идентификатор страницы сайта из таблицы {site}
lang_id	tinyint(2)	номер языковой версии сайта, с которой было отправлено

		сообщение
text	text	текст
admin_id	int(11)	пользователь, добавивший или первый отредактировавший сообщение в административной части, из таблицы {users}
user_id	int(11)	пользователь, добавивший сообщение из формы на сайте, из таблицы {users}
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{feedback_param}

Таблица модуля «Обратная связь». Хранит поля из конструктора сообщений

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
nameНОМЕР_ЯЗЫКА	varchar(250)	название
type	varchar(20)	тип поля
site_id	int(11)	идентификатор страницы сайта, на которой поле выводится, из таблицы {site}
sort	int(11)	порядковый номер для сортировки
required	enum('0','1')	обязательно для заполнения: 0 - нет, 1 - да
textНОМЕР_ЯЗЫКА	text	описание
config	text	настройки поля
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{feedback_param_element}

Таблица модуля «Обратная связь». Хранит значения полей из конструктора сообщений при добавлении сообщений

Поле	Тип (размерность)	Описание
------	-------------------	----------

id	int(11)	идентификатор
value	text	поле
param_id	int(11)	номер поля из таблицы {feedback_param}
element_id	int(11)	идентификатор сообщения из таблицы {feedback}
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{feedback_param_select}

Таблица модуля Обратная связь. Хранит название для полей типа select из конструктора сообщений

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
param_id	int(11)	номер поля из таблицы {feedback_param}
value	tinyint(2)	значение для типа характеристики «галочка»: 0 - нет, 1 - да
nameНОМЕР_ЯЗЫКА	varchar(50)	поле
sort	int(11)	порядковый номер для сортировки
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{files}

Таблица модуля «Файлы». Хранит информацию о страницах с файлами

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
nameНОМЕР_ЯЗЫКА	text	название
actНОМЕР_ЯЗЫКА	enum('0','1')	показывать на сайте: 0 - нет, 1 - да
date_start	int(10)	дата начала показа

date_finish	int(10)	дата окончания показа
map_no_show	enum('0','1')	не показывать на карте сайта: 0 - нет, 1 - да
changefreq	enum('always', 'hourly', 'daily', 'weekly', 'monthly', 'yearly', 'never')	Changefreq
priority	varchar(3)	Priority
cat_id	int(11)	идентификатор категории из таблицы {files_category}
site_id	int(11)	идентификатор страницы сайта из таблицы {site}
keywordsНОМЕР_ЯЗЫКА	varchar(250)	ключевые слова, тэг Keywords
descrНОМЕР_ЯЗЫКА	text	описание, тэг Description
title_metaНОМЕР_ЯЗЫКА	varchar(250)	заголовок окна в браузере, тэг Title
anonsНОМЕР_ЯЗЫКА	text	анонс
textНОМЕР_ЯЗЫКА	text	описание
link	varchar(255)	ссылка на внешний файл
sort	int(11)	порядковый номер для сортировки
timeedit	int(12)	время последнего изменения в формате UNIXTIME
access	enum('0', '1')	доступ ограничен: 0 - нет, 1 - да
admin_id	int(11)	пользователь, добавивший или первый отредактировавший файл в административной части, из таблицы {users}
theme	varchar(50)	шаблон страницы сайта
view	varchar(50)	шаблон модуля

trash

enum('0','1')

запись удалена в корзину: 0 - нет, 1 - да

{files_category}

Таблица модуля «Файлы». Хранит категории файлов (если они подключены)

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
nameНОМЕР_ЯЗЫКА	text	название
actНОМЕР_ЯЗЫКА	enum('0','1')	показывать на сайте: 0 - нет, 1 - да
map_no_show	enum('0','1')	не показывать на карте сайта: 0 - нет, 1 - да
changefreq	enum('always', 'hourly', 'daily', 'weekly', 'monthly', 'yearly', 'never')	Changefreq
priority	varchar(3)	Priority
parent_id	int(11)	идентификатор категории-родителя из таблицы {files_category}
count_children	tinyint(3)	количество вложенных категорий
site_id	int(11)	идентификатор страницы сайта из таблицы {site}
keywordsНОМЕР_ЯЗЫКА	varchar(250)	ключевые слова, тэг Keywords
descrНОМЕР_ЯЗЫКА	text	описание, тэг Description
title_metaНОМЕР_ЯЗЫКА	varchar(250)	заголовок окна в браузере, тэг Title
anonsНОМЕР_ЯЗЫКА	text	анонс
textНОМЕР_ЯЗЫКА	text	описание
sort	int(11)	порядковый номер для сортировки

timeedit	int(12)	время последнего изменения в формате UNIXTIME
access	enum('0', '1')	доступ ограничен: 0 - нет, 1 - да
admin_id	int(11)	пользователь, добавивший или первый отредктировавший категорию в административной части, из таблицы {users}
theme	varchar(50)	шаблон страницы сайта
view	varchar(50)	шаблон модуля
view_element	varchar(50)	шаблон модуля для элементов в категории
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{files_category_parents}

Таблица модуля «Файлы». Хранит информацию о вложенностях категорий друг в друга (если они подключены)

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
element_id	int(11)	идентификатор категории из таблицы {files_category}
parent_id	int(11)	идентификатор категории-родителя из таблицы {files_category}
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{files_category_rel}

Таблица модуля «Файлы». Хранит информацию о принадлежности файлов категориям (если они подключены)

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
element_id	int(11)	идентификатор файла из таблицы {files}
cat_id	int(11)	идентификатор категории из таблицы {files_category}

trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да
-------	---------------	---

{files_counter}

Таблица модуля «Файловых архив». Хранит информацию о просмотрах файлов

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
element_id	int(11)	идентификатор объявления из таблицы {files}
count_view	smallint(11)	количество просмотров
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{files_rel}

Таблица модуля «Файлы». Хранит информацию о «похожих» файлах

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
element_id	int(11)	идентификатор файла из таблицы {files}
rel_element_id	int(11)	идентификатор похожего файла из таблицы {files}
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{forum}

Таблица модуля «Форум». Хранит сообщения форума

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
author	int(11)	пользователь, добавивший сообщение, из таблицы {users}
parent_id	int(11)	идентификатор сообщения-родителя из таблицы {forum}
count_children	tinyint(3)	количество вложенных записей

created	int(10)	дата создания
date_update	int(12)	дата обновления
user_update	int(11)	пользователь, обновивший сообщение, из таблицы {users}
cat_id	int(11)	идентификатор темы форума из таблицы {forum_category}
text	text	сообщение
del	enum('0','1')	сообщение удалено модератором: 0 - да, 1 - нет
act	enum('0','1')	показывать на сайте: 0 - нет, 1 - да
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{forum_category}

Таблица модуля «Форум». Хранит категории форума

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
author	int(11)	пользователь, добавивший тему, из таблицы {users}
parent_id	int(11)	идентификатор категории-родителя из таблицы {forum_category}
count_children	tinyint(3)	количество вложенных записей
created	int(10)	дата создания
name	varchar(250)	название
date_update	int(12)	дата обновления
user_update	int(11)	пользователь, обновивший тему, из таблицы {users}
message_update	int(11)	дата последнего добавления/обновления темы
counter_view	int(11)	количество просмотров
del	enum('0','1')	тема удалена модератором: 0 - нет, 1 - да

act	enum('0','1')	показывать на сайте: 0 - нет, 1 - да
prior	enum('0','1')	тема закреплена: 0 - нет, 1 - да
close	enum('0','1')	тема закрыта: 0 - нет, 1 - да
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{forum_category_parents}

Таблица модуля «Форум». Хранит информацию о вложенностях категорий друг в друга

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
element_id	int(11)	идентификатор категории из таблицы {forum_category}
parent_id	int(11)	идентификатор категории-родителя из таблицы {forum_category}
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{forum_parents}

Таблица модуля «Форум». Хранит информацию о вложенностях сообщений друг в друга (сообщения на сообщения)

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
element_id	int(11)	идентификатор сообщения из таблицы {forum}
parent_id	int(11)	идентификатор сообщения-родителя из таблицы {forum}
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{forum_show}

Таблица модуля «Форум». Хранит информацию о новых сообщениях и темах для пользователей

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
element_id	int(11)	идентификатор сообщения из таблицы {forum}
user_id	int(11)	идентификатор пользователя из таблицы {users}
created	int(10)	дата создания
table_name	enum('forum','forum_category')	тип информации: forum - сообщение, forum_category - тема

{images}

Таблица модуля «Изображения». Хранит информацию о загруженных изображениях

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
name	varchar(250)	название
module_name	varchar(10)	название модуля, к которому прикреплено изображение
altНОМЕР_ЯЗЫКА	varchar(250)	атрибут alt
titleНОМЕР_ЯЗЫКА	varchar(250)	атрибут title
element_id	int(11)	идентификатор элемента модуля, к которому прикреплено изображение
size	int(11)	размер изображения
sort	int(11)	порядковый номер для сортировки
tmpcode	varchar(32)	временный идентификатор, если изображение прикрепляется к еще не созданному элементу
created	int(10)	дата создания
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{images_editor_folders}

Таблица модуля «Изображения». Хранит информацию о папках в плагине «Изображения» для визуального редактора

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
name	varchar(250)	название
parent_id	int(11)	идентификатор папки-родителя из таблицы {images_editor_folders}

{images_variations}

Таблица модуля «Изображения». Хранит варианты изображений

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
name	varchar(250)	название
folder	varchar(20)	название папки
quality	tinyint(2)	качество для изображений в формат JPEG
param	text	информация о применяемых действиях
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{keywords}

Таблица модуля «Перелинковка». Хранит информацию о ключевых словах и ссылках на них

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
text	text	ключевое слово
link	text	ссылка

actHOMEP_ЯЗЫКА	enum('0','1')	активен на сайте: 0 - нет, 1 - да
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{languages}

Таблица системного модуля «Языки сайта». Хранит информацию о существующих языковых версиях на сайте

Поле	Тип (размерность)	Описание
id	tinyint(2)	идентификатор
name	varchar(100)	название
shortname	varchar(10)	сокращенное название
base_admin	enum('0','1')	язык является основным для административной части: 0 - нет, 1 - да
base_site	enum('0','1')	язык является основным для пользовательской части: 0 - нет, 1 - да

{languages_translate}

Таблица ядра. Хранит перевод интерфейса

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
text	text	исходный текст
text_translate	text	перевод
lang_id	tinyint(2)	идентификатор языка из таблицы {languages}
module_name	varchar(50)	название модуля, для которого применим перевод
type	enum('admin', 'site')	часть сайта: admin - административная, site - пользовательская

{log}

Таблица ядра. Хранит последние полчаса неудачных попыток авторизации

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
ip	varchar(20)	IP-адрес пользователя
created	int(10)	дата в формате UNIXTIME
count	int(11)	количество неудачных попыток
info	text	дополнительная информация

{log_note}

Таблица ядра. Лог голосования

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
module_name	varchar(10)	название модуля
element_id	int(11)	идентификатор элемента модуля
note	int(7)	оценка
created	int(10)	дата создания
ip	varchar(62)	IP-адрес пользователя
session_id	varchar(64)	идентификатор сессии, поле session_id из таблицы {sessions}
include_name	varchar(10)	подключенный модуль

{menu}

Таблица модуля «Меню». Хранит пункты меню на сайте

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор

nameHOMEP_ЯЗЫКА	varchar(100)	название
module_name	varchar(13)	название модуля
site_id	int(11)	идентификатор страницы сайта из таблицы {site}
element_id	int(11)	идентификатор элемента модуля, если не задан module_cat_id
module_cat_id	int(11)	идентификатор категории модуля, если не задан element_id
cat_id	int(11)	идентификатор категории меню из таблицы {menu_category}
parent_id	int(11)	идентификатор пункта меню - родителя из таблицы {menu}
count_children	tinyint(3)	количество вложенных пунктов
othurl	varchar(250)	внешняя ссылка
access	enum('0','1')	доступ ограничен: 0 - нет, 1 - да
date_start	int(10)	дата начала показа
date_finish	int(10)	дата окончания показа
sort	int(11)	порядковый номер для сортировки
actHOMEP_ЯЗЫКА	enum('0','1')	показывать на сайте: 0 - нет, 1 - да
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{menu_category}

Таблица модуля «Меню». Хранит информацию о меню

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
nameHOMEP_ЯЗЫКА	varchar(100)	название

show_title	enum('0','1')	показывать заголовок: 0 - нет, 1 - да
show_all_level	enum('0','1')	выводить сразу все уровни вложенности: 0 - нет, 1 - да
hide_parent_link	enum('0','1')	не выводить ссылку, если есть вложенные пункты: 0 - нет, 1 - да
actHOMEP_ЯЗЫКА	enum('0','1')	показывать на сайте: 0 - нет, 1 - да
access	enum('0','1')	доступ ограничен: 0 - нет, 1 - да
current_link	enum('0','1')	текущий пункт ссылкой: 0 - нет, 1 - да
only_image	enum('0','1')	не отображать имя пункта меню, если используется изображение: 0 - нет, 1 - да
menu_template	varchar(100)	шаблон вывода меню
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{menu_parents}

Таблица модуля «Меню». Хранит информацию о вложенностях меню друг в друга

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
element_id	int(11)	идентификатор пункта меню из таблицы {menu}
parent_id	int(11)	идентификатор пункта-родителя из таблицы {menu}
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{messages}

Таблица модуля «Личные сообщения». Хранит личные сообщения пользователей

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
author	int(11)	пользователь, создавший сообщение, из таблицы {users}

to_user	int(11)	пользователь, которому адресовано сообщение, из таблицы {users}
text	text	текст сообщения
created	int(10)	дата создания
readed	enum('0','1')	сообщение прочитано: 0 - нет, 1 - да

{messages_user}

Таблица модуля «Личные сообщения». Хранит контакты пользователей

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
user_id	int(11)	пользователь, для которого формируется список контактов, из таблицы {users}
contact_user_id	int(11)	пользователь, с которым установлен контакт, из таблицы {users}
date_update	int(10)	дата последнего сообщения в контакте
readed	enum('0','1')	есть непрочитанные сообщения: 0 - да, 1 - нет
count_message	smallint(5)	количество сообщений

{mistakes}

Таблица модуля «Ошибки на сайте». Хранит личные сообщения об ошибках

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
url	varchar(255)	ссылка на сайте
created	int(10)	дата создания
selected_text	text	выделенный текст с ошибкой
comment	text	комментарий пользователя

{modules}

Таблица ядра. Хранит установленные модули

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
name	varchar(30)	название подмодуля
module_name	varchar(30)	название модуля
site	enum('0','1')	модуль имеет пользовательскую часть: 0 - нет, 1 - да
site_page	enum('0','1')	модуль можно прикрепить к старинице сайта: 0 - нет, 1 - да
admin	enum('0','1')	модуль имеет административную часть: 0 - нет, 1 - да
title	varchar(100)	заголовок модуля на русском языке

{news}

Таблица модуля «Новости». Хранит новости

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
nameНОМЕР_ЯЗЫКА	text	название
actНОМЕР_ЯЗЫКА	enum('0','1')	показывать на сайте: 0 - нет, 1 - да
map_no_show	enum('0','1')	не показывать на карте сайта: 0 - нет, 1 - да
changefreq	enum('always', 'hourly', 'daily', 'weekly', 'monthly', 'yearly', 'never')	Changefreq
priority	varchar(3)	Priority
created	int(10)	дата создания
date_start	int(10)	дата начала показа

date_finish	int(10)	дата окончания показа
cat_id	int(11)	идентификатор категории из таблицы {news_category}
site_id	int(11)	идентификатор страницы сайта из таблицы {site}
keywordsHOMEP_ЯЗЫКА	varchar(250)	ключевые слова, тэг Keywords
descrHOMEP_ЯЗЫКА	text	описание, тэг Description
title_metaHOMEP_ЯЗЫКА	varchar(250)	заголовок окна в браузере, тэг Title
anonsHOMEP_ЯЗЫКА	text	анонс
textHOMEP_ЯЗЫКА	text	описание
prior	enum('0','1')	важно, всегда сверху: 0 - нет, 1 - да
timeedit	int(12)	время последнего изменения в формате UNIXTIME
access	enum('0', '1')	доступ ограничен: 0 - нет, 1 - да
admin_id	int(11)	пользователь, добавивший или первый отредактировавший новость в административной части, из таблицы {users}
theme	varchar(50)	шаблон страницы сайта
view	varchar(50)	шаблон модуля
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{news_category}

Таблица модуля «Новости». Хранит категории новостей (если они подключены)

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор

nameHOMEP_ЯЗЫКА	text	название
actHOMEP_ЯЗЫКА	enum('0','1')	показывать на сайте: 0 - нет, 1 - да
map_no_show	enum('0','1')	не показывать на карте сайта: 0 - нет, 1 - да
changefreq	enum('always', 'hourly', 'daily', 'weekly', 'monthly', 'yearly', 'never')	Changefreq
priority	varchar(3)	Priority
parent_id	int(11)	идентификатор категории-родителя из таблицы {news_category}
count_children	tinyint(3)	количество вложенных категорий
site_id	int(11)	идентификатор страницы сайта из таблицы {site}
keywordsHOMEP_ЯЗЫКА	varchar(250)	ключевые слова, тэг Keywords
descrHOMEP_ЯЗЫКА	text	описание, тэг Description
title_metaHOMEP_ЯЗЫКА	varchar(250)	заголовок окна в браузере, тэг Title
anonsHOMEP_ЯЗЫКА	text	анонс
textHOMEP_ЯЗЫКА	text	описание
sort	int(11)	порядковый номер для сортировки
timeedit	int(12)	время последнего изменения в формате UNIXTIME
access	enum('0', '1')	доступ ограничен: 0 - нет, 1 - да
admin_id	int(11)	пользователь, добавивший или первый отредктировавший категорию в административной части, из таблицы {users}
theme	varchar(50)	шаблон страницы сайта

view	varchar(50)	шаблон модуля
view_element	varchar(50)	шаблон модуля для элементов в категории
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{news_category_parents}

Таблица модуля «Новости». Хранит информацию о вложенностях категорий новостей друг в друга (если они подключены)

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
element_id	int(11)	идентификатор категории из таблицы {news_category}
parent_id	int(11)	идентификатор категории-родителя из таблицы {news_category}
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{news_category_rel}

Таблица модуля «Новости». Хранит информацию о принадлежности новостей категориям (если они подключены)

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
element_id	int(11)	идентификатор новости из таблицы {news}
cat_id	int(11)	идентификатор категории из таблицы {news_category}
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{news_counter}

Таблица модуля «Новости». Хранит информацию о просмотрах новостей

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор

element_id	int(11)	идентификатор объявления из таблицы {news}
count_view	smallint(11)	количество просмотров
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{news_rel}

Таблица модуля «Новости».Хранит информацию о «похожих» новостях

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
element_id	int(11)	идентификатор новости из таблицы {news}
rel_element_id	int(11)	идентификатор похожей новости из таблицы {news}
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{photo}

Таблица модуля «Фотографии». Хранит фотографии

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
nameНОМЕР_ЯЗЫКА	text	название
actНОМЕР_ЯЗЫКА	enum('0','1')	показывать на сайте: 0 - нет, 1 - да
date_start	int(10)	дата начала показа
date_finish	int(10)	дата окончания показа
map_no_show	enum('0','1')	не показывать на карте сайта: 0 - нет, 1 - да
changefreq	enum('always', 'hourly', 'daily', 'weekly', 'monthly', 'yearly', 'never')	Changefreq
priority	varchar(3)	Priority

cat_id	int(11)	идентификатор категории из таблицы {photo_category}
site_id	int(11)	идентификатор страницы сайта из таблицы {site}
keywordsНОМЕР_ЯЗЫКА	varchar(250)	ключевые слова, тэг Keywords
descrНОМЕР_ЯЗЫКА	text	описание, тэг Description
title_metaНОМЕР_ЯЗЫКА	varchar(250)	заголовок окна в браузере, тэг Title
anonsНОМЕР_ЯЗЫКА	text	анонс
textНОМЕР_ЯЗЫКА	text	описание
sort	int(11)	порядковый номер для сортировки
timeedit	int(12)	время последнего изменения в формате UNIXTIME
access	enum('0', '1')	доступ ограничен: 0 - нет, 1 - да
admin_id	int(11)	пользователь, добавивший или первый отредактировавший фотографию в административной части, из таблицы {users}
theme	varchar(50)	шаблон страницы сайта
view	varchar(50)	шаблон модуля
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

[{photo_category}](#)

Таблица модуля «Фотографии». Хранит категории фотографий (если они подключены)

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
nameНОМЕР_ЯЗЫКА	text	название

actHOMEP_ЯЗЫКА	enum('0','1')	показывать на сайте: 0 - нет, 1 - да
map_no_show	enum('0','1')	не показывать на карте сайта: 0 - нет, 1 - да
changefreq	enum('always', 'hourly', 'daily', 'weekly', 'monthly', 'yearly', 'never')	Changefreq
priority	varchar(3)	Priority
parent_id	int(11)	идентификатор категории-родителя из таблицы {photo_category}
count_children	tinyint(3)	количество вложенных категорий
site_id	int(11)	идентификатор страницы сайта из таблицы {site}
keywordsHOMEP_ЯЗЫКА	varchar(250)	ключевые слова, тэг Keywords
descrHOMEP_ЯЗЫКА	text	описание, тэг Description
title_metaHOMEP_ЯЗЫКА	varchar(250)	заголовок окна в браузере, тэг Title
anonsHOMEP_ЯЗЫКА	text	анонс
textHOMEP_ЯЗЫКА	text	описание
sort	int(11)	порядковый номер для сортировки
timeedit	int(12)	время последнего изменения в формате UNIXTIME
access	enum('0', '1')	доступ ограничен: 0 - нет, 1 - да
admin_id	int(11)	пользователь, добавивший или первый отредктировавший категорию в административной части, из таблицы {users}
theme	varchar(50)	шаблон страницы сайта
view	varchar(50)	шаблон модуля

view_element	varchar(50)	шаблон модуля для элементов в категории
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{photo_category_parents}

Таблица модуля «Фотографии». Хранит информацию о вложенностях категорий фотографий друг в друга (если они подключены)

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
element_id	int(11)	идентификатор категории из таблицы {photo_category}
parent_id	int(11)	идентификатор категории-родителя из таблицы {photo_category}
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{photo_category_rel}

Таблица модуля «Фотографии». Хранит информацию о принадлежности фотографий категориям (если они подключены)

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
element_id	int(11)	идентификатор фотографии из таблицы {photo}
cat_id	int(11)	идентификатор категории из таблицы {photo_category}
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{photo_counter}

Таблица модуля «Фотогалерея». Хранит информацию о просмотрах фотографий

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
element_id	int(11)	идентификатор объявления из таблицы {photo}

count_view	smallint(11)	количество просмотров
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{photo_rel}

Таблица модуля «Фотогалерея». Хранит информацию о «похожих» фотографиях

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
element_id	int(11)	идентификатор фотографии из таблицы {photo}
rel_element_id	int(11)	идентификатор похожей фотографии из таблицы {photo}
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{rating}

Таблица модуля «Рейтинг»

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
element_id	int(11)	идентификатор элемента модуля
module_name	varchar(17)	название модуля
rating	float	средний рейтинг
count_votes	int(11)	количество проголосовавших
created	int(10)	дата последнего голосования
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{redirect}

Таблица ядра. Хранит редиректы на страницы сайта, элементы и категории модулей

Поле	Тип (размерность)	Описание
------	-------------------	----------

id	int(11)	идентификатор
redirect	varchar(255)	страница, с которой стоит редирект
code	smallint(5)	код редиректа
module_name	varchar(8)	название модуля
element_id	int(11)	идентификатор элемента модуля, на который стоит редирект
site_id	int(11)	идентификатор страницы сайта из таблицы {site} , на которую стоит редирект
cat_id	int(11)	идентификатор категории, на которую стоит редирект
param_id	int(11)	номер поля из таблицы {*_param} , на которое стоит редирект
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{referrals}

Таблица модуля «Коды РК». Хранит рефералы

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
created	int(10)	дата создания
referer	text	откуда пришел пользователь

{rewrite}

Таблица ядра. Хранит информацию о всех ЧПУ на сайте

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
rewrite	varchar(250)	ЧПУ
module_name	varchar(8)	название модуля

element_id	int(11)	идентификатор элемента модуля
site_id	int(11)	идентификатор страницы сайта из таблицы {site}
cat_id	int(11)	идентификатор категории
param_id	int(11)	номер поля из таблицы {*_param}
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{search_history}

Таблица модуля «Поиск». Хранит историю поиска

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
created	int(10)	дата запроса
name	text	поисковый запрос

{search_index}

Таблица модуля «Поиск». Хранит поисковый индекс

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
keyword_id	int(11)	идентикатор слова из таблицы {search_keywords}
result_id	int(11)	идентификатор страницы из таблицы {search_results}

{search_keywords}

Таблица модуля «Поиск». Хранит слова

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
keyword	varchar(255)	слово

{search_results}

Таблица модуля «Поиск». Хранит проиндексированные страницы

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
name	varchar(255)	название
url	varchar(255)	адрес страницы
snippet	varchar(255)	сниппет
element_id	int(11)	номер элемента модуля
table_name	varchar(50)	название проиндексированной таблицы
lang_id	tinyint(2)	идентификатор языка сайта из таблицы {languages}
rating	tinyint(2)	рейтинг страницы для сортировки результатов (определяется рейтингом модуля)

{sessions}

Таблица ядра. Хранит сессии авторизованных пользователей

Поле	Тип (размерность)	Описание
user_id	int(11)	идентификатор пользователя из таблицы {users}
session_id	varchar(64)	идентификатор сессии
hostname	varchar(128)	хост
user_agent	varchar(255)	браузер пользователя
timestamp	varchar(20)	время, до которого сессия действует
session	text	сериализованные данные сессии

{sessions_hash}

Таблица ядра. Хранит контрольные хеши авторизованных пользователей

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
user_id	int(11)	идентификатор пользователя из таблицы {users}
type	enum('user','admin')	часть сайта: user - пользовательская, admin - административная
hash	char(32)	хеш
created	int(10)	дата создания

{shop}

Таблица модуля «Магазин». Хранит товары

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
nameНОМЕР_ЯЗЫКА	text	название
actНОМЕР_ЯЗЫКА	enum('0','1')	показывать на сайте: 0 - нет, 1 - да
article	varchar(30)	артикул
date_start	int(10)	дата начала показа
date_finish	int(10)	дата окончания показа
map_no_show	enum('0','1')	не показывать на карте сайта: 0 - нет, 1 - да
changefreq	enum('always', 'hourly', 'daily', 'weekly', 'monthly', 'yearly', 'never')	Changefreq
priority	varchar(3)	Priority
cat_id	int(11)	идентификатор категории из таблицы {shop_category}
site_id	int(11)	идентификатор страницы сайта из таблицы

[{site}](#)

keywords	НОМЕР_ЯЗЫКА	varchar(250)	ключевые слова, тэг Keywords
descr	НОМЕР_ЯЗЫКА	text	описание, тэг Description
title_meta	НОМЕР_ЯЗЫКА	varchar(250)	заголовок окна в браузере, тэг Title
anons	НОМЕР_ЯЗЫКА	text	анонс
text	НОМЕР_ЯЗЫКА	text	описание
yandex		text	данные для выгрузки в Яндекс.Маркет
show_yandex		enum('0','1')	выгружать в Яндекс.Маркет: 0 - нет, 1 - да (если в настройках выбрана выгрузка только выбранных товаров)
no_buy		enum('0','1')	товар временно отсутствует: 0 - нет, 1 - да
import		enum('0','1')	товар только что импортирован: 0 - нет, 1 - да
import_id		varchar(32)	собственный идентификатор товара при импорте
sort		int(11)	порядковый номер для сортировки
timeedit		int(12)	время последнего изменения в формате UNIXTIME
counter_buy		int(11)	количество покупок
hit		enum('0', '1')	хит: 0 - нет, 1 - да
new		enum('0', '1')	новинка: 0 - нет, 1 - да
action		enum('0', '1')	акция: 0 - нет, 1 - да
is_file		enum('0', '1')	это товар-файл: 0 - нет, 1 - да
access		enum('0', '1')	доступ ограничен: 0 - нет, 1 - да

admin_id	int(11)	пользователь, добавивший или первый отредактировавший товар в административной части, из таблицы {users}
theme	varchar(50)	шаблон страницы сайта
view	varchar(50)	шаблон модуля
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{shop_additional_cost}

Таблица модуля «Магазин». Хранит сопутствующие услуги

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
nameНОМЕР_ЯЗЫКА	text	название
textНОМЕР_ЯЗЫКА	text	описание
actНОМЕР_ЯЗЫКА	enum('0','1')	показывать на сайте: 0 - нет, 1 - да
percent	float	процент от стоимости товаров в корзине
price	float	цена
amount	float	бесплатно от
required	enum('0','1')	всегда включено в цену: 0 - нет, 1 - да
sort	int(11)	порядковый номер для сортировки
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{shop_cart}

Таблица модуля «Магазин». Хранит содержимое корзины авторизованных пользователей

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор

good_id	int(11)	идентификатор товара из таблицы {shop}
user_id	int(11)	идентификатор пользователя из таблицы {users}
price_id	int(11)	идентификатор цены товара - поле price_id из таблицы {shop_price}
created	int(10)	дата добавления в корзину
param	text	сериализованные данные о характеристиках товара (доступных к выбору при заказе)
is_file	enum('0','1')	товар-файл: 0 - нет, 1 - да
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{shop_category}

Таблица модуля «Магазин». Хранит категории магазина

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
nameНОМЕР_ЯЗЫКА	text	название
actНОМЕР_ЯЗЫКА	enum('0','1')	показывать на сайте: 0 - нет, 1 - да
map_no_show	enum('0','1')	не показывать на карте сайта: 0 - нет, 1 - да
changefreq	enum('always', 'hourly', 'daily', 'weekly', 'monthly', 'yearly', 'never')	Changefreq
priority	varchar(3)	Priority
parent_id	int(11)	идентификатор категории-родителя из таблицы {shop_category}
count_children	tinyint(3)	количество вложенных категорий
site_id	int(11)	идентификатор страницы сайта из таблицы {site}

keywordsHOMEP_ЯЗЫКА	varchar(250)	ключевые слова, тэг Keywords
descrHOMEP_ЯЗЫКА	text	описание, тэг Description
title_metaHOMEP_ЯЗЫКА	varchar(250)	заголовок окна в браузере, тэг Title
anonsHOMEP_ЯЗЫКА	text	анонс
textHOMEP_ЯЗЫКА	text	описание
show_yandex	enum('0','1')	выгружать в Яндекс.Маркет: 0 - нет, 1 - да (если в настройках выбрана выгрузка категорий только выбранных товаров)
import	enum('0','1')	категория только что импортирована: 0 - нет, 1 - да
import_id	varchar(32)	собственный идентификатор категории при импорте
sort	int(11)	порядковый номер для сортировки
timeedit	int(12)	время последнего изменения в формате UNIXTIME
access	enum('0', '1')	доступ ограничен: 0 - нет, 1 - да
admin_id	int(11)	пользователь, добавивший или первый отредактировавший категорию в административной части, из таблицы {users}
theme	varchar(50)	шаблон страницы сайта
view	varchar(50)	шаблон модуля
view_element	varchar(50)	шаблон модуля для элементов в категории
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{shop_category_parents}

Таблица модуля «Магазин». Хранит информацию о вложенностях категорий магазина друг в друга

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
element_id	int(11)	идентификатор категории из таблицы {shop_category}
parent_id	int(11)	идентификатор категории-родителя из таблицы {shop_category}
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{shop_category_rel}

Таблица модуля «Магазин». Хранит информацию о принадлежности товаров к категориям

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
element_id	int(11)	идентификатор товара из таблицы {shop}
cat_id	int(11)	идентификатор категории из таблицы {shop_category}
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{shop_counter}

Таблица модуля «Каталог товаров». Хранит информацию о просмотрах товара

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
element_id	int(11)	идентификатор объявления из таблицы {shop}
count_view	smallint(11)	количество просмотров
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{shop_currency}

Таблица модуля «Магазин». Хранит валюты сайта

Поле	Тип (размерность)	Описание
------	-------------------	----------

id	smallint(5)	идентификатор
name	varchar(20)	название
exchange_rate	float	курс к основной валюте
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{shop_delivery}

Таблица модуля «Магазин». Хранит способы доставки

Поле	Тип (размерность)	Описание
id	smallint(5)	идентификатор
name	varchar(20)	название
description	text	описание
active	enum('0','1')	показывать на сайте: 0 - нет, 1 - да
sort	int(11)	порядковый номер для сортировки
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{shop_delivery_thresholds}

Таблица модуля «Магазин». Хранит пороги значений стоимости способов доставки

Поле	Тип (размерность)	Описание
id	smallint(5)	идентификатор
delivery_id	smallint(5)	идентификатор способа доставки из таблицы {shop_delivery}
price	float	стоимость
amount	float	от
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{shop_discount}

Таблица модуля «Магазин». Хранит скидки

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
date_start	int(11)	дата начала действия
date_finish	int(11)	дата окончания действия
discount	float	процент скидки
amount	float	действует от цены товара
deduction	float	фиксированная сумма скидки
threshold	int(11)	действует от общей суммы заказа
threshold_cumulative	int(11)	действует от общей оплаченных заказов
role_id	int(11)	тип пользователя, для которого установлена скидка, из таблицы {users_role}
act	enum('0','1')	показывать на сайте: 0 - нет, 1 - да
coupon	varchar(10)	код купона
user_id	int(11)	пользователь, для которого установлена скидка, из таблицы {users}
count_use	tinyint(3)	оставшееся количество возможностей использовать купон при покупке
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{shop_discount_object}

Таблица модуля «Магазин». Хранит информацию об объектах на скидки (товары или категории)

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
discount_id	int(11)	идентификатор скидки из таблицы {shop_discount}

cat_id	int(11)	идентификатор категории из таблицы {shop_category}
good_id	int(11)	идентификатор товара из таблицы {shop}
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{shop_files_codes}

Таблица модуля «Магазин». Хранит коды для скачивания товаров-файлов

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
shop_id	int(11)	идентификатор товара из таблицы {shop}
code	varchar(50)	код для скачивания товара-файла
date_finish	datetime	дата и время окончания действия кода

{shop_import}

Таблица модуля «Магазин». Хранит описания полей файла для импорта/экспорта

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
name	varchar(250)	название
cat_id	int(11)	идентификатор категории импорта из таблицы {shop_import_category}
type	varchar(10)	тип
params	text	сериализованные данные о поле
required	enum('0','1')	выдавать ошибку, если поле не заполнено: 0 - нет, 1 - да
sort	int(11)	порядковый номер для сортировки
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{shop_import_category}

Таблица модуля «Магазин». Хранит описание файлов импорта/экспорта

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
name	varchar(250)	название
format	enum('csv','xls')	формат файла
type	enum('good','category')	тип данных: good - товары, category - категории товаров
delete_items	enum('0','1')	удалять не описанные в файле данные: 0 - нет, 1 - да
site_id	int(11)	идентификатор страницы сайта из таблицы {site}
cat_id	int(11)	идентификатор категории товаров из таблицы {shop_category}
count_part	smallint(5)	количество строк, выгружаемых за один проход скрипта
delimiter	varchar(20)	разделитель данных в строке
end_string	varchar(20)	обозначать конец строки символом
encoding	varchar(20)	кодировка
sub_delimiter	varchar(20)	разделитель данных внутри поля
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{shop_order}

Таблица модуля «Магазин». Хранит информацию о заказах

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
user_id	int(11)	идентификатор пользователя из таблицы {users}
created	int(10)	дата создания
status	enum('0','1','2','3','4')	действие статуса заказа
status_id	tinyint(3)	идентификатор статуса заказа из таблицы

[{shop_order_status}](#)

lang_id	tinyint(2)	идентификатор языковой версии сайта, с которой был сделан заказ, из таблицы {languages}
summ	float	общая сумма заказа
payment_id	smallint(5)	метод оплаты заказа из таблицы {shop_payment}
delivery_id	smallint(5)	способ доставки из таблицы {shop_delivery}
delivery_summ	float	стоимость доставки
code	varchar(32)	уникальный код заказа для просмотра квитанций и накладных
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{shop_order_additional_cost}

Таблица модуля «Магазин». Хранит информацию о сопутствующих услугах в заказах

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
order_id	int(11)	идентификатор заказа из таблицы {shop_order}
additional_cost_id	smallint(5)	идентификатор сопутствующей услуги из таблицы {shop_additional_cost}
summ	float	стоимость сопутствующей услуги

{shop_order_goods}

Таблица модуля «Магазин». Хранит информацию о товарах в заказах

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
order_id	int(11)	идентификатор заказа из таблицы {shop_order}
good_id	int(11)	идентификатор товара из таблицы {shop}

discount_id	int(11)	идентификатор скидки из таблицы {shop_discount}
count_goods	int(5)	количество товаров
price	float	цена
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{shop_order_goods_param}

Таблица модуля «Магазин». Хранит информацию о характеристиках товаров в заказах

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
value	int(11)	значение параметра - идентификатор из таблицы {shop_param_select}
param_id	int(11)	идентификатор характеристики товара из таблицы {shop_param}
order_goods_id	int(11)	идентификатор купленного товара из таблицы {shop_order_goods}

{shop_order_param}

Таблица модуля «Магазин». Хранит поля из конструктора форм при оформлении заказа в корзине

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
nameНОМЕР_ЗАКАЗА	varchar(250)	поле
textНОМЕР_ЯЗЫКА	text	описание
type	varchar(20)	тип поля
sort	int(11)	порядковый номер для сортировки
required	enum('0','1')	обязательно для заполнения: 0 - нет, 1 - да
show_in_form_register	enum('0','1')	выводить в форме редактирования данных

config	text	сериализованные параметры поля
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{shop_order_param_element}

Таблица модуля «Магазин». Хранит информацию о заполненных полях при оформлении заказа (адрес доставки и т.д.)

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
value	text	значение
param_id	int(11)	номер поля из таблицы {shop_order_param}
element_id	int(11)	идентификатор заказа из таблицы {shop_order}
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{shop_order_param_select}

Таблица модуля «Магазин». Хранит значения полей из конструктора формы оформления заказа

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
param_id	int(11)	номер поля из таблицы {shop_order_param}
value	tinyint(2)	значение для типа характеристики «галочка»: 0 - нет, 1 - да
nameНОМЕР_ЯЗЫКА	varchar(50)	поле
sort	int(11)	порядковый номер для сортировки
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{shop_order_param_user}

Таблица модуля «Магазин

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
value	text	поле
param_id	int(11)	номер поля из таблицы {shop_order_param}
user_id	int(11)	идентификатор пользователя из таблицы {users}
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{shop_order_status}

Таблица модуля «Магазин». Хранит статусы заказа

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
nameНОМЕР_ЯЗЫКА	varchar(50)	поле
status	enum('0','1','2','3','4')	действие
sort	int(11)	порядковый номер для сортировки
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{shop_param}

Таблица модуля «Магазин». Хранит названия дополнительных характеристик

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
nameНОМЕР_ЯЗЫКА	varchar(250)	название
type	varchar(20)	тип
sort	int(11)	порядковый номер для сортировки
search	enum('0','1')	выводить в форме поиска: 0 - нет, 1 - да

list	enum('0','1')	выводить в списке: 0 - нет, 1 - да
block	enum('0','1')	выводить в блоке товаров: 0 - нет, 1 - да
required	enum('0','1')	обязательно для заполнения: 0 - нет, 1 - да
page	enum('0','1')	отдельная страница для значений: 0 - нет, 1 - да
textНОМЕР_ЯЗЫКА	text	описание
config	text	сериализованные настройки характеристики
display_in_sort	enum('0','1')	выводить в блоке сортировки: 0 - нет, 1 - да
measure_unitНОМЕР_ЯЗЫКА	varchar(50)	единица измерения
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{shop_param_category_rel}

Таблица модуля «Магазин». Хранит информацию о принадлежности характеристики к категории магазина

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
element_id	int(11)	идентификатор характеристики из таблицы {shop_param}
cat_id	int(11)	идентификатор категории товара из таблицы {shop_category}
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{shop_param_element}

Таблица модуля «Магазин». Хранит значения дополнительных характеристик

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
valueНОМЕР_ЯЗЫКА	text	значение

param_id	int(11)	номер поля из таблицы {shop_param}
element_id	int(11)	идентификатор товара из таблицы {shop}
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{shop_param_select}

Таблица модуля «Магазин». Хранит значения дополнительных характеристик с типом «выпадающий список» и «галочка»

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
param_id	int(11)	номер поля из таблицы {shop_param}
value	tinyint(2)	значение для типа характеристики «галочка»: 0 - нет, 1 - да
nameНОМЕР_ЯЗЫКА	varchar(50)	значение для списка
sort	int(11)	порядковый номер для сортировки
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{shop_payment}

Таблица модуля «Магазин». Хранит способы электронной оплаты

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
nameНОМЕР_ЯЗЫКА	varchar(50)	название
textНОМЕР_ЯЗЫКА	text	описание
actНОМЕР_ЯЗЫКА	enum('0','1')	показывать на сайте: 0 - нет, 1 - да
payment	varchar(20)	платежная система
sort	int(11)	порядковый номер для сортировки

trash

enum('0','1')

запись удалена в корзину: 0 - нет, 1 - да

{shop_pay_history}

Таблица модуля «Магазин». Хранит историю платежей

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
order_id	int(11)	идентификатор заказа из таблицы {shop_order}
status	enum('request_pay', 'pay')	статус: request_pay - запрос платежа, pay - платеж
user_id	int(11)	идентификатор пользователя из таблицы {users}
created	int(10)	дата платежа
summ	float	сумма
payment	varchar(50)	платежная система

{shop_price}

Таблица модуля «Магазин». Хранит информацию обо всех ценах на товары

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
good_id	int(11)	идентификатор товара из таблицы {shop}
price	float	цена
old_price	float	исходная цена
count_goods	int(11)	количество товара
price_id	int(11)	идентификатор исходной цены из таблицы {shop_price}
date_start	int(10)	дата начала действия
date_finish	int(10)	дата окончания действия

discount	float	скидка
discount_id	int(11)	идентификатор скидки из таблицы {shop_discount}
threshold	int(11)	действует от общей суммы заказа
threshold_cumulative	int(11)	действует от общей оплаченных заказов
user_id	int(11)	идентификатор пользователя, для которого действует скидка, из таблицы {users}
role_id	int(11)	идентификатор типа пользователя, для которого действует скидка, из таблицы {users_role}
currency_id	smallint(5)	идентификатор валюты из таблицы {shop_currency}
import_id	varchar(32)	собственный идентификатор при импорте
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{shop_price_image_rel}

Таблица модуля «Магазин». Хранит информацию об изображениях, прикрепленных к цене

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
price_id	int(11)	поле price_id из таблицы {shop_price}
image_id	int(11)	идентификатор изображения из таблицы {images}
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{shop_price_param}

Таблица модуля «Магазин». Хранит информацию характеристиках товара, на которые распространяются цены

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
price_id	int(11)	поле price_id из таблицы {shop_price}

param_id	int(11)	идентификатор характеристики из таблицы {shop_param}
param_values	int(11)	значение характеристики - идентификатор из таблицы {shop_param_select}
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{shop_rel}

Таблица модуля «Магазин». Хранит информацию о «похожих» товарах

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
element_id	int(11)	идентификатор товара из таблицы {shop}
rel_element_id	int(11)	идентификатор похожего товара из таблицы {shop}
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{shop_waitlist}

Таблица модуля «Магазин». Хранит список ожиданий

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
good_id	int(11)	идентификатор товара из таблицы {shop}
user_id	int(11)	идентификатор пользователя из таблицы {users}
lang_id	tinyint(2)	идентификатор языка сайта из таблицы {languages}
mail	varchar(64)	e-mail пользователя
created	int(10)	дата добавления товара в список ожиданий
param	text	сериализованные данные о характеристиках товара
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{shop_wishlist}

Таблица модуля «Магазин». Хранит список пожеланий

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
good_id	int(11)	идентификатор товара из таблицы {shop}
user_id	int(11)	идентификатор пользователя из таблицы {users}
session_id	int(11)	идентификатор сессии - поле session_id из таблицы {sessions}
created	int(10)	дата добавления товара в список ожиданий
count	smallint(5)	количество товара
param	text	сериализованные данные о характеристиках товара
is_file	enum('0','1')	это товар-файл: 0 - нет, 1 - да
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{site}

Таблица ядра. Хранит основное дерево страниц сайта

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
parent_id	int(11)	идентификатор страницы-родителя из таблицы {site}
count_children	tinyint(3)	количество вложенных страниц
nameНОМЕР_ЯЗЫКА	varchar(100)	название
titleНОМЕР_ЯЗЫКА	varchar(250)	заголовок
title_metaНОМЕР_ЯЗЫКА	varchar(250)	заголовок окна в браузере, тэг Title
keywordsНОМЕР_ЯЗЫКА	varchar(250)	ключевые слова, тэг Keywords

descrHOMEP_ЯЗЫК A	text	описание, тэг Description
textHOMEP_ЯЗЫКА	longtext	контент
actHOMEP_ЯЗЫКА	enum('0','1')	показывать на сайте: 0 - нет, 1 - да
access	enum('0','1')	доступ к странице ограничен: 0 - нет, 1 - да
date_start	int(10)	дата начала показа
date_finish	int(10)	дата окончания показа
admin_id	int(11)	пользователь, добавивший или первый отредактировавший страницу в административной части, из таблицы {users}
title_no_show	enum('0','1')	не копировать заголовок в H1: 0 - нет, 1 - да
map_no_show	enum('0','1')	не показывать на карте сайта: 0 - нет, 1 - да
changefreq	enum('always', 'hourly', 'daily', 'weekly', 'monthly', 'yearly', 'never')	Changefreq
priority	varchar(3)	Priority
noindex	enum('0','1')	не индексировать: 0 - нет, 1 - да
search_no_show	enum('0','1')	не участвует в поисковой выдаче: 0 - нет, 1 - да
block	enum('0','1')	блок на сайте: 0 - нет, 1 - да
sort	int(11)	порядковый номер для сортировки
timeedit	int(12)	время последнего изменения в формате UNIXTIME
theme	varchar(50)	шаблон страницы
module_name	varchar(20)	прикрепленный модуль
js	text	JS-код
is_menu	enum('0','1')	страница ваыведена в меню: 0 - нет, 1 - да

trash

enum('0','1')

запись удалена в корзину: 0 - нет, 1 - да

{site_block_rel}

Таблица ядра. Хранит информацию на каких страницах размещается блок информации

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
element_id	int(11)	идентификатор страницы-блока из таблицы {site}
site_id	int(11)	идентификатор страницы, к которой прикреплен блок, из таблицы {site}
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{site_parents}

Таблица ядра. Хранит информацию о вложенностях страниц сайта друг в друга

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
element_id	int(11)	идентификатор страницы из таблицы {site}
parent_id	int(11)	идентификатор страницы-родителя из таблицы {site}
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{subscription}

Таблица модуля «Рассылки». Хранит рассылки

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
created	int(10)	дата создания
send	enum('0','1')	рассылка была отправлена: 0 - нет, 1 - да
cat_id	int(11)	идентификатор категории из таблицы {subscription_category}

name	varchar(250)	название
text	text	содержание рассылки
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{subscription_category}

Таблица модуля «Рассылки». Хранит категории рассылок

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
nameНОМЕР_ЯЗЫКА	varchar(250)	название
act	enum('0','1')	показывать на сайте: 0 - нет, 1 - да
sort	int(11)	порядковый номер для сортировки
textНОМЕР_ЯЗЫКА	text	описание
parent_id	int(11)	идентификатор категории-родителя из таблицы {subscription_category}
count_children	tinyint(3)	количество вложенных категорий
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{subscription_category_parents}

Таблица модуля «Рассылки». Хранит информацию о вложенностях категорий друг в друга

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
element_id	int(11)	идентификатор категории из таблицы {subscription_category}
parent_id	int(11)	идентификатор категории-родителя из таблицы {subscription_category}
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{subscription_category_rel}

Таблица модуля «Рассылки». Хранит информацию о принадлежности рассылок категориям

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
element_id	int(11)	идентификатор рассылки из таблицы {subscription}
cat_id	int(11)	идентификатор категории из таблицы {subscription_category}
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{subscription_emails}

Таблица модуля «Рассылки». Хранит электронные адреса пользователей

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
mail	varchar(64)	электронный ящик
code	varchar(32)	уникальный код для управления подпиской
created	int(10)	дата создания
act	enum('0','1')	активен: 0 - нет, 1 - да
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{subscription_emails_unrel}

Таблица модуля «Рассылки». Хранит категории рассылок, на которые ящик не подписан

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
element_id	int(11)	идентификатор электронного адреса из таблицы {subscription_category}

cat_id	int(11)	идентификатор категории из таблицы {subscription_category}
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{subscription_phones}

Таблица модуля «Рассылки». Хранит телефоны пользователей

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
phone	varchar(64)	телефон
created	int(10)	дата создания
act	enum('0','1')	активен: 0 - нет, 1 - да
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{subscription_sms}

Таблица модуля «Рассылки». Хранит SMS-рассылки

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
created	int(10)	дата создания
send	enum('0','1')	рассылка была отправлена: 0 - нет, 1 - да
name	varchar(250)	название
text	text	содержание рассылки
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{tags}

Таблица модуля «Теги». Хранит информацию о прикрепленных тегах к разным объектам сайта

Поле	Тип (размерность)	Описание
------	-------------------	----------

id	int(11)	идентификатор
module_name	varchar(10)	название модуля, к которому прикреплен тег
element_id	int(11)	идентификатор элемента модуля, к которому прикреплен тег
tags_name_id	tinyint(3)	идентификатор тега из таблицы {tags_name}
actHOMEP_ЯЗЫКА	enum('0','1')	показывать на сайте: 0 - нет, 1 - да
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{tags_name}

Таблица модуля Теги. Хранит теги

Поле	Тип (размерность)	Описание
id	int(11)	поле
nameHOMEP_ЯЗЫКА	varchar(70)	название
map_no_show	enum('0','1')	не показывать на карте сайта: 0 - нет, 1 - да
changefreq	enum('always', 'hourly', 'daily', 'weekly', 'monthly', 'yearly', 'never')	Changefreq
priority	varchar(3)	Priority
sort	int(11)	порядковый номер для сортировки
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{trash}

Таблица ядра. Хранит данные об удаленных в корзину элементах

Поле	Тип	Описание
------	-----	----------

(размерность)

id	int(11)	идентификатор
table_name	varchar(50)	таблица, в которых храниться удаленный элемент
module_name	varchar(13)	название модуля, которому принадлежит удаленный элемент
element_id	int(11)	идентификатор удаленного элемента модуля
created	int(10)	дата удаления
parent_id	int(11)	идентификатор записи, удаление которой повлекло удаление текущего элемента, из таблицы {trash}
count_children	tinyint(3)	количество вложенных записей
user_id	int(11)	идентификатор пользователя из таблицы {users}

{trash_parents}

Таблица ядра. Хранит данные о взаимосвязи удаления записей

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
element_id	int(11)	идентификатор записи из таблицы {trash}
parent_id	int(11)	идентификатор записи-родителя из таблицы {trash}

{update}

Таблица ядра. Хранит информацию о файлах, закрытых для обновления

Поле	Тип (размерность)	Описание
id	tinyint(3)	поле
user_id	int(11)	идентификатор пользователя из таблицы {users}
name	text	путь до файла

created int(11) дата создания

text text комментарий

{users}

Таблица модуля «Пользователи». Хранит пользователей сайта

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
name	varchar(60)	логин
password	varchar(32)	пароль
mail	varchar(64)	e-mail
created	int(10)	дата создания
fio	varchar(250)	ФИО
role_id	int(11)	идентификатор типа пользователя из таблицы {users_role}
act	enum('0','1')	активен на сайте: 0 - нет, 1 - да
htmleditor	enum('0','1')	использовать визуальный редактор: 0 - нет, 1 - да
lang_id	tinyint(3)	идентификатор языка сайта из таблицы {languages}
background	varchar(255)	обои для административной части
admin_nastr	smallint(5)	количество элементов на странице в административной части
identity	varchar(255)	URL на страницу в соц. сети
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{users_actlink}

Таблица модуля «Пользователи». Код активации аккаунта

Поле	Тип (размерность)	Описание
------	-------------------	----------

id	int(11)	идентификатор
link	varchar(250)	код активации
user_id	int(11)	идентификатор ользователя из таблицы {users}
created	int(10)	дата создания

{users_param}

Таблица модуля «Пользователи». Поля конструктора форы регистрации

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
nameНОМЕР_ЯЗЫКА	varchar(250)	поле
textНОМЕР_ЯЗЫКА	text	описание
type	varchar(20)	тип поля
show_in_page	enum('0','1')	выводить на странице пользователя: 0 - нет, 1 - да
show_in_form_auth	enum('0','1')	выводить в форме редактирования данных для авторизованных пользователей: 0 - нет, 1 - да
show_in_form_no_auth	enum('0','1')	выводить в форме регистрации: 0 - нет, 1 - да
sort	int(11)	порядковый номер для сортировки
required	enum('0','1')	обязательно для заполнения: 0 - нет, 1 - да
config	text	сериализованные данные о настройках поля
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{users_param_element}

Таблица модуля «Пользователи». Заполненная информация из конструктора формы регистрации о пользователе

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
value	text	значение
param_id	int(11)	номер поля из таблицы {users_param}
element_id	int(11)	идентификатор пользователя из таблицы {users}
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{users_param_role_rel}

Таблица модуля «Пользователи». Хранит информацию о применимости поля конструктора к типу пользователя

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
element_id	int(11)	номер поля из таблицы {users_param}
role_id	tinyint(3)	идентификатор типа пользователя из таблицы {users_role}
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{users_param_select}

Таблица модуля «Пользователи». Хранит значения полей с типом «выпадающий список» и «галочка»

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
param_id	int(11)	номер поля из таблицы {users_param}
value	tinyint(2)	значение для типа характеристики «галочка»: 0 - нет, 1 - да
nameНОМЕР_ЯЗЫКА	varchar(50)	название

sort	int(11)	порядковый номер для сортировки
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{users_role}

Таблица модуля «Пользователи». Типы пользователей сайта

Поле	Тип (размерность)	Описание
id	int(3)	поле
nameНОМЕР_ЯЗЫКА	varchar(64)	название
registration	enum('0','1')	задавать при регистрации: 0 - нет, 1 - да
only_self	enum('0','1')	видеть только свои материалы: 0 - нет, 1 - да
actНОМЕР_ЯЗЫКА	enum('0','1')	активен на сайте: 0 - нет, 1 - да
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{users_role_perm}

Таблица модуля «Пользователи». Хранит информацию о правах типов пользователей

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
role_id	tinyint(3)	идентификатор типа пользователя из таблицы {users_role}
perm	text	права на действия: all, init, edit, del
rewrite	text	тег доступа (например, название модуля)
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{votes}

Таблица модуля «Опросы». Хранит варианты ответа

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор

nameНОМЕР_ЯЗЫКА	text	ответ
actНОМЕР_ЯЗЫКА	enum('0','1')	показывать на сайте: 0 - нет, 1 - да
cat_id	int(11)	идентификатор вопроса из таблицы {votes_category}
count_votes	int(5)	количество ответов
sort	int(11)	порядковый номер для сортировки
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{votes_category}

Таблица модуля «Опросы». Хранит вопросы

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
nameНОМЕР_ЯЗЫКА	text	вопрос
actНОМЕР_ЯЗЫКА	enum('0','1')	показывать на сайте: 0 - нет, 1 - да
no_result	enum('0','1')	не показывать результаты: 0 - нет, 1 - да
userversion	enum('0','1')	пользователи могут дать свой вариант ответа: 0 - нет, 1 - да
sort	int(11)	порядковый номер для сортировки
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{votes_category_site_rel}

Таблица модуля «Опросы». Хранит информацию о страницах сайта, на которых выводятся опросы

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор

element_id	int(11)	идентификатор вопроса из таблицы {votes_category}
site_id	int(11)	идентификатор страницы сайта из таблицы {site}
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

{votes_userversion}

Таблица модуля «Опросы». Хранит варианты ответов пользователей

Поле	Тип (размерность)	Описание
id	int(11)	идентификатор
cat_id	int(11)	идентификатор вопроса из таблицы {votes_category}
text	text	текст
trash	enum('0','1')	запись удалена в корзину: 0 - нет, 1 - да

Устройство модуля

Модули в diafan.CMS основаны на [MVC](#) паттерне. Это означает, что в одной части модуля данные готовятся и обрабатываются, в другой части эти данные оформляются и выводятся на сайт. Управляется все это контроллером. Сделано это для того, чтобы можно было как угодно править оформление модуля, не переживая за алгоритм обработки данных, и наоборот. То есть, каждый разработчик может переоформить внешний вид модуля новостей не затрагивая алгоритм. Модуль должен обязательно иметь 3 части:

- [контроллер](#) – файл *modules/имя_модуля/имя_модуля.php* класс *Имя_модуля*;
- [модель](#) – файл *modules/имя_модуля/имя_модуля.model.php* класс *Имя_модуля_model*;
- [шаблон](#) – файлы в папке *modules/имя_модуля/views*, формат файла *modules/имя_модуля/views/имя_модуля.view.имя_шаблона.php*.

Пример:

Модуль «Новости» состоит из файла *modules/news/news.php* – это контроллер модуля. Там описаны все переменные, используемые модулем, описаны все функции, которые выполняются в модели и т.д. *modules/news/news.model.php* – модель модуля, в которой работают функции модуля. В модели запрашиваются данные из БД, идет их обработка и готовится переменная-массив *\$result* для передачи информации в шаблон модуля. Файлы *modules/news/views/...* – шаблон модуля. В шаблоне принимается переменная *\$result* из модели модуля и все данные из нее оформляются и выводятся на сайт.

Внимание! Если нужно внести изменения в алгоритм работы любого модуля, необходимо править файл модель *modules/имя_модуля/имя_модуля.model.php*. А если нужно поправить отображение элементов модуля на сайте, то необходимо править файл-шаблон *modules/имя_модуля/имя_модуля.view.имя_шаблона.php*.

Если в модуле обрабатываются данных из формы, отправленные с помощью Ajax, то добавляется файл:

- [обработка](#) – файл *modules/имя_модуля/имя_модуля.ajax.php* класс *Имя_модуля_ajax*.

Если модуль имеет часть «Подключение», то есть используется в других модулях, то добавляется еще один файл:

- [подключение](#) – файл *modules/имя_модуля/имя_модуля.inc.php* класс *Имя_модуля_inc*.

Если модуль устанавливается опционально, то должен существовать файл инсталляции:

- [инсталляция](#) – файл *modules/имя_модуля/имя_модуля.install.php*.

Если страница модуля должна существовать независимо от структуры сайта, то можно обратиться к файлу модуля напрямую. Для этого файл должен иметь название *modules/модуль/модуль.функция.php*. Обратиться к нему можно так: *http://site.ru/модуль/функция/*.

- [прямое обращение](#) – файл *modules/модуль/модуль.функция.php*.

Модуль может [взаимодействовать с другими модулями](#). Для этого нужно подготовить файлы в формате, предусмотренном сторонними модулями.

Контроллер

- [Класс Controller](#)

Контроллер модуля – класс *Модуль* в файле *modules/модуль/модуль.php*.

В классе описаны следующие функции:

- `init` – инициализация модуля;
- `show_module` – вывод контента модуля;
- шаблонные функции.

Функция инициализации модуля вызывается, если модуль подключен к текущей странице. В ней происходит:

1. Определение переменных, передаваемых в URL, которые используются в модуле.

Пример:

```
$this->rewrite_variable_names = array('page', 'show', 'cat', 'sort',  
'param');  
$this->diafan->rewrite_variable_names = $this->rewrite_variable_names;
```

2. Подключение модели и выбор функции модели модуля в зависимости от переданных параметров.

Пример:

```
// создаем экземпляр объекта модели модуля «Магазин»  
$model = new Shop_model($this->diafan);  
// если в URL передана переменная show, вызываем функцию id в модели –  
// показываем страницу товара  
if($this->diafan->show)  
{  
    $this->result = $model->id();  
}  
// иначе вызываем функцию list_ в модели – выводим список товаров  
else  
{  
    $this->result = $model->list_();  
}
```

3. Определение глобальных переменных.

Глобальные переменные модуля – заголовок страницы, навигация «Хлебные крошки», время редактирования страницы, ключевые слова, описание страницы, спрятать ссылки на предыдущую, последующую страницы. Подробнее смотрите в [разделе «Архитектура diafan.CMS»](#).

Пример:

```
$this->get_global_variables();
```

В примере используется стандартная функция из класса *Controller*. Вместо этого можно использовать свой алгоритм.

Функция вывода контента модуля вызывается из общей [шаблонной функции show_body](#). Её основное назначение – в зависимости от переданных параметров вызвать функцию в шаблоне модуля.

Функции шаблона вызываются с помощью подключения шаблонов.

Пример:

```
// если в URL передана переменная show, подключаем файл
modules/shop/views/shop.view.id.php – показываем страницу товара
if($this->diafan->show)
{
    $this->diafan->_tpl->get('id', 'shop', $this->result);
}
// иначе подключаем файл modules/shop/views/shop.view.list.php – выводим список
товаров
else
{
    $this->diafan->_tpl->get('list', 'shop', $this->result);
}
```

Данные, сгенерированные в модели, передаем в шаблон с помощью свойства класса *\$result*.

Шаблонные функции модуля – определяются условия, при которых функция не отображается, фильтруются полученные атрибуты, подключаются соответствующие функции в модели и представлении модуля. Подробнее смотрите в [разделе «Как добавить шаблонный тег»](#).

Класс Controller

Контроллер модуля наследует класс *Controller* – это каркас для всех страниц.

Свойства

array **\$rewrite_variable_names**

- var: переменные, передаваемые в URL страницы
- access: public

array **\$result**

- var: сгенерированные в модели данные, передаваемые в шаблон

- access: protected

string **\$view**

- var: представление модуля для текущей страницы
- access: protected

Методы

void **init ()** – инициализация модуля.

- access: public

void **show_module ()** – выводит шаблон модуля.

- access: public

boolean true **get_global_variables ()** – определяет свойства страницы, если они заданы в модуле.

- access: protected

array **get_attributes (array \$attributes)** – задает неопределенным атрибутам шаблонного тега значение по умолчанию.

- access: protected
- array \$attributes: массив определенных атрибутов

Модель

- [Класс Model](#)
- [Методы](#)

Модель модуля – класс *Модуль_model* в файле *modules/модуль/модуль.model.php*.

Содержание файла может быть различным в зависимости от заложенного в модуле функционала. Как правило, класс модели описывает следующие функции:

- [list\(\)](#) – формирует список элементов;
- [list_category\(\)](#) – формирует список элементов в категории;
- [first_page\(\)](#) – первая страница модуля;
- [id\(\)](#) – формирует данные для страницы элемента;
- [show_block\(\)](#) – формирует данные для шаблонной функции.

Функции отдадут данные в виде массива, который затем передается в шаблон.

Для ускорения генерирования страницы данные кэшируются. Подробнее о кэшировании в diafan.CMS смотрите [раздел «Кэширование»](#).

Класс Model

Модель модуля наследует класс *Model*, описанный в файле *includes/model.php*.

Методы

boolean **access** (integer *\$element_id*, [integer *\$cat_id* = 0], [string *\$module* = ""]) – проверяет, есть ли доступ к элементу/категории модуля.

- access: protected
- integer *\$element_id*: номер элемента
- integer *\$cat_id*: номер категории
- string *\$module_name*: модуль

Если при вызове опустить аргумент *\$element_id*, то проверка будет вызвана для текущего модуля. Проверка прав доступна только для авторизованных пользователей и для страниц с настроенным полем *Доступ*.

Внимание! О настройке прав доступа более подробно можно узнать в разделе ["Права доступа"](#).

Пример:

```
/* modules/clauses/clauses.model.php  
проверка прав доступа в модуле «Статьи» */  
if (empty($row) || (!empty($row['access']) && !$this->access(0, $row['id'])))
```

```
{
    include_once(ABSOLUTE_PATH.'includes/403.php');
}
```

string **format_date** (integer \$date, [string \$module_name = "], [integer \$site_id = 0]) – форматирует дату в соответствии с конфигурацией модуля.

- access: protected
- integer \$date: дата в [формате UNIX](#)
- string \$module_name: название модуля, по умолчанию модуль, прикрепленный к текущей странице
- integer \$site_id: номер страницы сайта

Пример:

```
/* modules/clauses/clauses.model.php
конвертация из UNIX-формата даты создания статьи в модуле «Статьи» */
$this->result["date"] = $this->format_date($row["created"]);
```

array **get_author** (integer \$author) – получает имя, никнейм и аватар пользователя сайта.

- access: protected
- integer \$author: номер пользователя сайта

При вызове функции производится запрос к таблице БД *users*, где по значению переменной *\$author* ищется соответствующий ей id пользователя. В случае неудачи возвращается «Гость».

Пример:

```
/* modules/comments/comments.ajax.php
запрос данных автора комментария по его id */
if ($this->diafan->configmodules('user_name', 'comments') && $row["user_id"])
{
    $row["name"] = $this->get_author($row["user_id"]);
}
```

string **get_error** ([string \$module = "], [string \$field = "]) – выдает ошибку о неправильном заполнении формы.

- access: protected
- string \$module: название модуля
- string \$field: название поля

Пример:

```
/* modules/faq/faq.model.php
проверка правильности заполнения полей формы при отправлении вопроса; модуль «Вопрос-ответ» */
$result["error_name"] = $this->get_error("faq".$site_id, 'name');
```

```
$result["error_question"] = $this->get_error("faq".$site_id, 'question');  
$result["error_email"] = $this->get_error("faq".$site_id, 'email');
```

array **get_params** (array \$config) – получает массив полей формы.

- access: protected
- array \$config: настройки функции: module модуль, table таблица, where условие для SQL-запроса

Используется в модулях, в которых есть возможность добавить новые поля при помощи [конструктора](#).

Пример:

```
/* modules/feedback/feedback.ajax.php  
получение дополнительных полей при обработке формы модуля «Обратная связь» */  
$params = $this->get_params(array("module" => "feedback", "where" =>  
"site_id=".$this->site_id));
```

array **get_path** (string \$module) – генерирует данные для навигации «Хлебные крошки».

- access: protected
- string \$module: модуль

Пример:

```
/* modules/ads/ads.model.php  
использования метода в модуле «Объявления» */  
$this->result["path"] = $this->get_path('ads');
```

boolean **validate_attribute_site_cat** (string \$module_name, array \$site_ids, array \$cat_ids) – валидация атрибутов cat_id и site_id для шаблонных тегов.

- access: protected
- string \$module_name: название модуля
- array \$site_id: страница сайта
- array \$cat_id: категория

Осуществляется проверка на использование нечисловых значений, а также на синтаксические ошибки при вызове шаблона с параметрами \$site_id или \$cat_id.

Пример:

```
/* modules/news/news.model.php  
проверка значений cat_id и site_id в модуле «Новости» */  
if(! $this->validate_attribute_site_cat('news', $site_ids, $cat_ids))  
{  
    return false;  
}
```

void **theme_view** () – определяет шаблоны страницы и модуля для элемента.

- access: protected

Пример:

```
/* modules/clauses/clauses.model.php  
широко используется в моделях при выводе списком без деления на категории */  
$this->theme_view();
```

void theme_view_first_page () – определяет шаблоны страницы и модуля для первой страницы модуля, если используются категории.

- access: protected

Пример:

```
/* modules/clauses/clauses.model.php  
Используется при выводе первой страницы (first page) */  
$this->theme_view_first_page();  
return $this->result;
```

void theme_view_cat (array \$row) – определяет шаблоны страницы и модуля для категории.

- access: protected
- array \$row: данные о текущей категории

Пример:

```
/* modules/clauses/clauses.model.php  
используется при выводе списка с категориями */  
$this->theme_view_cat($row);
```

void theme_view_element (array \$row) – определяет шаблоны страницы и модуля для элемента.

- access: protected
- array \$row: данные о текущем элементе

Пример:

```
/* modules/clauses/clauses.model.php  
используется при выводе страницы отдельного элемента */  
$this->theme_view_element($row);
```

void meta (array \$row) – определяет значения META-тегов элемента.

- access: protected
- array \$row: данные о текущем элементе

Пример:

```
/* modules/clauses/clauses.model.php  
используется при выводе со страницы отдельного элемента */  
$this->meta($row);
```

void meta_cat (array \$row) – определяет значения МЕТА-тегов категории.

- access: protected
- array \$row: данные о текущей категории

Пример:

```
/* modules/clauses/clauses.model.php  
используется при выводе со страницы списка с категориями */  
$this->meta_cat($row);
```

void counter_view () – счетчик просмотров элемента.

- access: protected

Функция собирает данные о просмотрах и хранит их в базе данных. Для сохранения статистики просмотров необходимо включить опцию «Счетчик просмотров» в настройках модуля.

Пример:

```
/* modules/clauses/clauses.model.php  
считает количество просмотров элемента */  
$this->counter_view();
```

Внимание!

Для хранения данных о просмотрах используется уникальная для каждого модуля таблица с постфиксом `_counter`. Например, `diafan_photo_counter` – для элементов фотогалереи.

Шаблон

Шаблон модуля – файлы в папке *modules/модуль/views*. Название файла должно быть *modules/модуль/views/модуль.view.шаблон.php*.

Обычно состав файлов следующий:

- *модуль.view.form.php* – выводит в пользовательской части форму (если модуль интерактивный). Например, в модуле «Вопрос-ответ» форму отправки вопроса;
- *модуль.view.list.php* – выводит в пользовательской части список элементов модуля. Например, в модуле «Новости» – список новостей;
- *модуль.view.first_page.php* – выводит главную страницу модуля;
- *модуль.view.id.php* – выводит в пользовательской части отдельный элемент. Например, отдельную новость в модуле «Новости»; или отдельный товар в модуле «Магазин»
- *модуль.view.show_block.php* – выводит шаблонную функцию модуля. Например, для модуля «Новости», вывод блока новостей для шаблонного тега [show_block](#).

В шаблоне могут быть другие файлы, отражающие индивидуальные особенности модуля.

Подключение шаблона из [контроллера](#) происходит следующим образом:

Пример:

```
$this->diafan->_tpl->get('название_шаблона', 'название_модуля', $result);
```

где **\$result** – передаваемые в шаблон данные в виде массива, полученные из [модели](#).

В шаблоне для вызова другого шаблона можно воспользоваться сокращенной формой:

Пример:

```
$this->get('название_шаблона', 'название_модуля', $result);
```

Пример:

```
/* вывод шаблона кнопки «Купить» shop.view.buy_form
из шаблона страницы товара shop.view.id */
$this->get('buy_form', 'shop', array("row" => $result, "result" => $result));
```

Обработка Ajax-запросов

- [Класс Ajax](#)
- [Методы](#)

Все данные из формы обрабатываются классом *Модуль_ajax*, описанным в файле *modules/модуль/модуль.ajax.php*.

В файле должна быть описана основная функция – *ajax_request()*, вызываемая при передаче данных модулю. Функция подключается всегда, когда передана переменная `$_POST["module"] = 'модуль';`

То есть данные модулю могут быть переданы даже если он не подключен ни к одной странице сайта. Уже в функции *ajax_request()* проверяется соответствие прикрепленного к странице модуля и вызванного модуля, если это необходимо.

Функция *ajax_request()* выдает ответ *true* или *false*, означающий, приняты ли данные и стоит ли прерывать загрузку страницы.

Структура класса может быть любой для того чтобы реализовывать предусмотренный функционал.

Класс Ajax

Класс *Модуль_ajax* наследует класс *Ajax*, описанный в файле *includes/ajax.php*.

Методы

boolean **check_captcha ()** – проверяет правильность каптчи.

- access: protected

При вызове функции она проверяет наличие подключения защитного кода в таблице БД *config*, после чего обращается к [методам класса «Captcha»](#) для проверки правильности ввода. Если каптча введена неверно, будет возвращено сообщение с ошибкой.

Пример:

```
/* modules/faq/faq.ajax.php  
используется при проверке отправляемой формы, например в модуле «Вопрос-ответ»  
*/  
$this->check_captcha();
```

void **empty_required_field (array \$config)** – проверка на заполнение обязательных полей.

- access: protected
- array \$config: настройки функции: params поля формы, prefix префикс

Если в форме указаны поля, требующие обязательного заполнения, то корректность ввода может быть проверена этой функцией. Список требующих проверки полей должен быть передан в виде аргумента.

Пример:

```
/* modules/registration/registration.ajax.php  
проверка ввода обязательных полей при регистрации */  
$this->empty_required_field(array("params" => $params));
```

void insert_values (array \$config) – добавляет значение полей формы в базу данных.

- access: protected
- array \$config: настройки функции: *id* номер элемента, *table* таблица, *params* поля формы

Данная функция получает список аргументов, в соответствии с которым производит запись значений в базу данных. Вызывается при обработке данных из формы.

Пример:

```
/* modules/cart/cart.ajax.php  
обработка данных при отправлении заказа в модуле «Корзина» */  
$this->insert_values(array("id" => $order_id, "table" => "shop_order", "params" => $params));
```

boolean send_errors () – отправляет ответ.

- access: protected

Метод отправляет ответ формата JSON, который будет выводиться при отправлении формы. Например, если при заполнении и отправлении формы не было заполнено обязательное поле «Имя», то вернется ответ с сообщением «Пожалуйста, заполните обязательное поле Ваше имя». В случае успешной отправки будет выведен текст, определенный в настройках модуля, например «Спасибо за ваше сообщение!»

Пример:

```
/* modules/ads/ads.ajax.php  
при обработке формы ошибок не возникло, в конце функции происходит возврат  
сообщения об успешной отправке; модуль «Объявления» */  
return $this->send_errors()
```

string to_json (array \$result) – преобразует массив в формат JSON.

- access: protected
- array \$result: исходный массив

Обрабатывать данные при ajax-запросах часто удобнее при их представлении в виде JSON-данных. В данной функции происходит преобразование массива в строку-представление JSON. Для работы используется сторонний файл */plugins/json.php*

Пример:

```
/* modules/cart/cart.ajax.php  
возврат данных из корзины после пересчета цены */  
echo to_json($json);
```

void update_values (array \$config) – обновляет значение полей формы в базу данных.

- access: protected
- array \$config: настройки функции: *id* номер элемента, *table* таблица, *params* поля формы, *prefix* префикс.

Пример:

```
/* modules/usersettings/usersettings.ajax.php  
изменение данных текущего пользования при редактировании профиля через личный кабинет */  
$this->update_values(array("id" => $this->diafan->_user->id, "table" => "users", "params" => $params));
```

boolean valid_email (string \$email, string \$field) – проверка e-mail на валидность.

- access: protected
- string \$email: e-mail
- string \$field: название поля в массиве \$_POST

Используется сторонний файл */includes/validate.php*. При неудачной проверке будет возвращено значение TRUE, и добавлено сообщение о неверном формате электронной почты.

Пример:

```
/* modules/faq/faq.ajax.php  
проверка правильности формата введенной электронной почты при отправке вопроса;  
модуль «Вопрос-ответ» */  
$this->valid_email($_POST['email'], "email");
```

Подключение

Для того чтобы использовать модуль в других модулях, в файле *modules/модуль/модуль.inc.php* описывается класс *Модуль_inc*.

Обращение к подключению модуля происходит через переменную `$this->diafan->_модуль`. При первом обращении к переменной создается экземпляр класса *Модуль_inc*.

Класс подключения наследует класс *Model*, описанный в файле *includes/model.php*. Его описание смотрите в [разделе «Модель»](#).

Для примера, давайте реализуем получение новости по её идентификатору из любой [модели](#) CMS.

Для этого создадим в папке *modules/news/* файл *news.inc.php*

Пример:

```
<?php
class News_inc extends Model
{
    /**
     * Получение новости
     * @param int $id идентификатор новости
     * @return array или null если новости нет
     */
    public function get($id)
    {
        $query=DB::query("SELECT [name],[anons] FROM {news} WHERE id=%d AND [act]='1'
        AND trash='0'", $id);
        if($query === false) return null;
        return DB::fetch_array($query);
    }
}
```

Проверим наш метод в файле *modules/shop/shop.model.php* в функции *get_result()*, которая возвращает результаты, сформированные в модели *shop*. Допишем:

Пример:

```
$this->result["news"]=$this->diafan->_news->get(1);
print_r($this->result["news"]);
exit;
```

Перейдя на страницу каталога, мы должны увидеть нашу новость под идентификатором 1.

Инсталляция

Для того чтобы модуль мог быть установлен опционально, создается файл инсталляции модуля. Это файл *modules/модуль/модуль.install.php*.

В файле определены следующие переменные и функции:

- `$title` – название модуля;
- `$db` – описание запросов к базе данных, необходимых для установки модуля;
- `$example` – массив SQL запросов для установки примеров заполнения модуля;
- `module_basic_модуль()` – выполняет действия для основной установки модуля;
- `module_example_модуль()` – выполняет действия для установки примеров заполнения модуля;
- `module_basic_uninstall_модуль()` – деинсталлирует модуль.

Формат массива `$db`:

Пример:

```
$db = array(
  // таблицы модуля
  "tables" => array(
    array(
      "name" => "название_таблицы",
      // поля таблицы
      "fields" => array(
        array(
          // пример: id
          "name" => "название_поля",
          // пример: INT(11) UNSIGNED NOT NULL AUTO_INCREMENT
          "type" => "описание_поля",
          // поле переводится на другие языки (true|false)
          "multilang" => true,
        ),
        ...
      ),
      // ключи таблицы
      "keys" => array(
        // пример PRIMARY KEY (id) или KEY site_id (`site_id`)
        "описание_ключа",
        ...
      ),
    ),
    ...
  ),
  // записи в таблицу {modules}
  "modules" => array(
    array(
      // пример: cart
      "name" => "название_подмодуля",
```

```

// пример: shop
"module_name" => "название_модуля",
// имеет административную часть (true|false)
"admin" => true,
// имеет пользовательскую часть (true|false)
"site" => true,
// можно прикрепить к странице сайта (true|false)
"site_page" => true,
// пример: Каталог товаров
"title" => "название_модуля_на_русском",
),
),
// страницы сайта
"sites" => array(
    array(
        // пример: Каталог товаров, Shop
        "name" => array("название_на_русском", "название_на_английском"),
        // активен на сайте (true|false)
        "act" => true,
        // пример: 15
        "sort" => "порядковый_номер_для_сортировки",
        // пример: shop – не обязательный параметр
        "module_name" => "прикрепленный_модуль",
        // пример: catalog-tovarov
        "rewrite" => "ЧПУ",
        // пример: sitemap.php – не обязательный параметр
        "theme" => "шаблон_страницы",
        // не показывать на карте сайта (true|false)
        "map_no_show" => true,
        // не индексировать (true|false)
        "noindex" => true,
        // не участвует в поисковой выдаче (true|false)
        "search_no_show" => true,
        // ссылки в меню – не обязательный параметр
        "menu" => array(
            "cat_id" => "номер_категории_меню",
            "sort" => "порядковый_номер_для_сортировки",
        ),
    ),
    ...
),
// меню административной части сайта
"adminsites" => array(
    array(
        "name" => "название",
        "rewrite" => "ЧПУ",
        "group_id" => "номер_группы",
        "sort" => "порядковый_номер_для_сортировки",
        // показывать в меню (true|false)
        "act" => true,
        "docs" => "ссылка на документацию",
        // вложенные страницы

```

```

        "children" => array(
            array(
                "name" => "название",
                "rewrite" => "ЧПУ",
                "sort" => "порядковый_номер_для_сортировки",
                // показывать в меню (true|false)
                "act" => true,
            ),
        ),
    ),
    ...
),
// настройки модуля по-умолчанию
"config" => array(
    array(
        "name" => "название",
        // по умолчанию, текущий модуль
        "module_name" => "название_модуля",
        // если значение переводятся на разные языки, то можно задать в виде массива
        "value" => "значение",
    ),
),
// SQL-запросы
"sql" => array(
    "SQL-запрос",
    ...
),
);

```

При установке модуля обязательно нужно сделать запрос к таблице [{modules}](#), чтобы система определила модуль как установленный.

Если у полей таблиц модуля предусмотрено различное содержание для различных языковых версий, то в SQL-запросе поле обрабатывается функцией *multilang()*. Данная функция повторяет код для каждой языковой версии, при этом заменяя код LANG на номер языка. Если функции переданы несколько аргументов, то для первого языка выводится первый аргумент, для второго – второй и т. д. Если число языков превышает количество аргументов, переданных функции, то для последующих языков сайта берется последний аргумент.

Пример:

В модуле «Магазин» название товаров переводится. Сайт переводится на 4 языка: русский, английский, немецкий, французский.

```

// запрос на создание примера заполнения таблицы товаров
$example[] = "INSERT INTO {shop} (id ".multilang(" ", nameLANG").")
VALUES (1".multilang(" ", 'Коршун'", " ", 'Vulture'", " ", "").").";";

```

В результате сформируются следующий SQL запрос к базе данных:


```
INSERT INTO diafan_shop (id, name1, name2, name3, name4) VALUES (  
1, 'Коршун', 'Vulture', "", "  
);
```

Файл инсталляции используется при установке diafan.CMS, при установке / удалении модуля из административной части, при восстановлении базы данных из административной части.

Прямое обращение к файлам модуля

Для работы вне структуры сайта описывается файл *modules/модуль/модуль.функция.php*. Этот файл можно вызвать напрямую так: *http://site.ru/модуль/функция/*.

Прямое обращение к модулю может понадобиться по следующим причинам:

1. Страница модуля должна существовать независимо от того, создана ли соответствующая страница в структуре сайта. Тем самым нужно обезопасить функциональность от случайного удаления страницы.
2. Страница модуля имеет служебный характер и не должна отображаться на карте сайта, в файле *sitemap.xml*, в меню и пр.
3. При работе с частью модуля нужно избавиться от лишних запросов на определение страницы.

В файле *modules/модуль/модуль.функция.php* доступен основной класс системы `$this->diafan`. Через эту переменную можно получить доступ к классам пользователей, кэширования, маршрутизатора и подключения шаблонов, а также к подключениям модулей.

Кроме того, перед подключением файлов происходит старт сессии, определяется запрошенная языковая версия сайта, временная зона сайта и пользователь. Все эти данные можно использовать в файле модуля.

Если запрошен адрес *http://site.ru/модуль/функция/какие-то_данные/*, то данные, переданные в URL, будут доступны в модуле в виде: `$_GET["rewrite"] = 'какие-то_данные';`

Примеры реализации прямого обращения к модулю в diafan.CMS:

- *modules/captcha/captcha.get.php* – *http://site.ru/captcha/get/* – генерирует изображение защитного кода;
- *modules/news/news.rss.php* – *http://site.ru/news/rss/* – генерирует RSS-ленту новостей;
- *modules/cart/cart.payment.php* – *http://site.ru/cart/payment/* – работа с платежными системами;
- *modules/useradmin/useradmin.edit.php* – *http://site.ru/useradmin/edit/* – форма редактирования данных из пользовательской части;
- *modules/attachments/attachments.get.php* – *http://site.ru/attachments/get/* – отдает прикрепленный файл.

Взаимодействие с другими модулями

Некоторые модули должны учитывать особенности других модулей в своей работе. Эти особенности должны быть описаны в файлах других модулей в нужном формате.

Например, формирование карты сайта (файл *sitemap.xml*) должно учитывать особенности построения ссылок во всех модулях. Поэтому в diafan.CMS есть файлы *modules/news/news.sitemap.php*, *modules/shop/shop.sitemap.php*, *modules/tags/tags.sitemap.php* и др.

Файл описывает класс *Модуль_sitemap* и функцию *get()* в нем.

`boolean true get (integer $site_id, integer $timeedit)` – генерирует карту модуля.

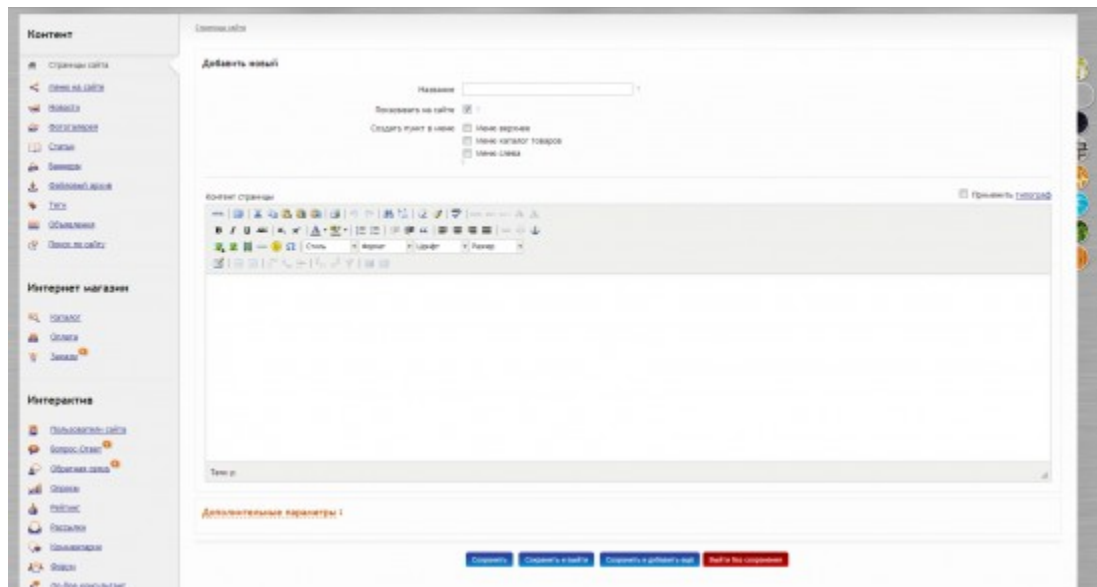
- access: public
- integer \$site_id: номер страницы сайта
- integer \$timeedit: время редактирования страницы

Формат файла заложен в модуле «Карта сайта».

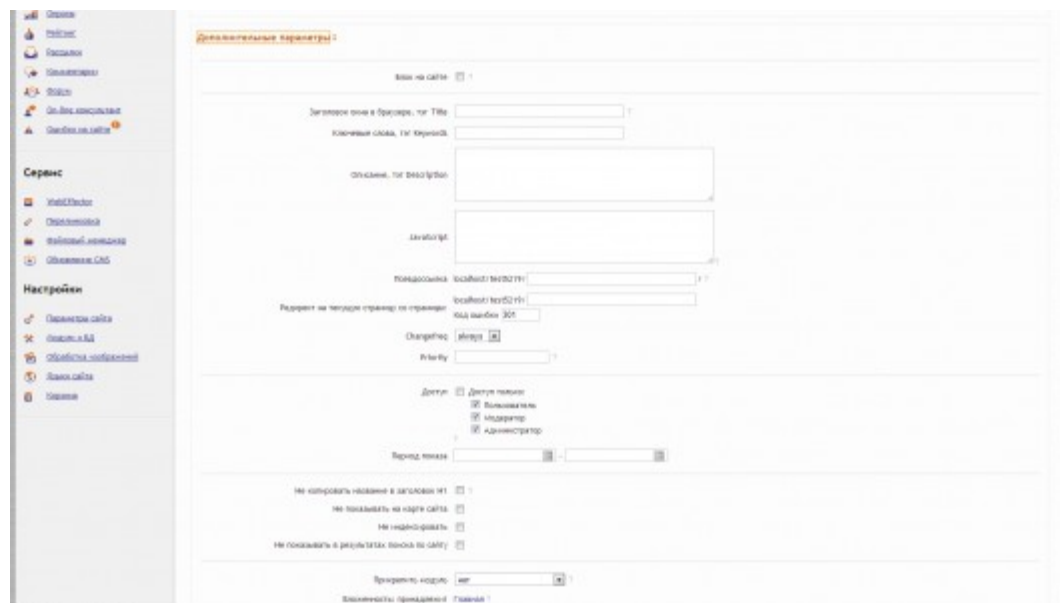
Подключение модуля

Подключаемый модуль выбирается из списка модулей при редактировании страницы сайта.

Шаг 1. Открываем страницу на редактирование



Шаг 2. Переходим в раздел «Дополнительные параметры»



Шаг 3. Выбираем модуль из выпадающего списка

Административная часть сайта

Если файл *index.php* определил, что запрошена административная часть, то подключается файл *adm/index.php*. Файл выполняет следующие функции:

1. подключение всех «жизненно важных» файлов;
2. инициация основного объекта системы.

Основной объект системы – экземпляр класса *Init_admin*, описанный в файле *adm/includes/init.php*.

При инициации основного объекта системы происходит:

1. старт сессии;
2. определение администратора;
3. определение запрошенной языковой версии для редактирования;
4. определение запрошенного модуля;
5. подключение [администрирования модуля](#);
6. подключение шаблона административной части.

Экземпляр класса *Init_admin* передается во все другие объекты и доступен из любого места системы в виде переменной `$this->diafan`.

Таким образом, в любом месте системы можно получить глобальные данные:

1. данные о текущей странице;
2. переданные в URL переменные;
3. общие функции;
4. объекты классов-подключений.

Теперь подробнее об этих данных:

Данные о текущей странице

В область глобальной видимости попадают следующие данные о текущей странице и данные, сформированные модулем:

- `$this->diafan->cid` – номер текущей страницы;
- `$this->diafan->rewrite` – ЧПУ текущей страницы сайта;
- `$this->diafan->name` – название текущей страницы;
- `$this->diafan->theme` – файл шаблона текущей страницы;
- `$this->diafan->text` – контент текущей страницы;
- `$this->diafan->parent_id` – номер страницы родителя;
- `$this->diafan->module` – модуль, прикрепленный к текущей странице.

Переданные в URL переменные

Список всех доступных переменных задан в массиве `$this->diafan->rewrite_variable_names`.

- `$this->diafan->page` – номер страницы;

- `$this->diafan->addnew` – значение 1 при добавлении элемента;
- `$this->diafan->edit` – номер элемента для редактирования;
- `$this->diafan->savenew` – значение 1 при сохранении нового элемента;
- `$this->diafan->save` – номер элемента для сохранения;
- `$this->diafan->cat` – номер категории;
- `$this->diafan->site` – номер страницы сайта;
- `$this->diafan->parent` – номер родителя;
- `$this->diafan->error` – код ошибки, уведомления.

Пример:

Страница редактирования товара – <http://www.site.ru/admin/shop/edit13/>

Общие функции

Класс *Init_admin* наследует абстрактный класс Core (*includes/core.php*). Все функции этого класса можно вызвать в виде:

`$this->diafan->название_функции();`

Пример:

```
/* adm/includes/rel_elements.php
обрезаем название ссылки до 50 символов методом short_text из класса Core */
echo '<a href="#">' . $this->diafan->short_text($row[$name], 50) . '</a>';
```

Объекты классов-подключений

В основном объекте иницируются классы:

- пользователи (класс *User_admin*, файл *includes/user.php*, вызывается `$this->diafan->_user`);
- маршрутизатор (класс *Route*, файл *includes/route.php*, вызывается `$this->diafan->_route`);
- подключение шаблонов (класс *Template*, файл *includes/template.php*, вызывается `$this->diafan->_tpl`).

Кроме того, через основной объект можно обратиться к любому модулю-подключению. Если в модуле есть файл *modules/модуль/модуль.inc.php*, где описан класс *Модуль_inc*, то при первом обращении к переменной `$this->diafan->_модуль`, создастся экземпляр класса.

Основные действия

- [Вывод данных в виде списка](#)
- [Редактирование, добавление](#)
- [Валидация](#)
- [Сохранение](#)
- [Удаление, удаление в корзину](#)
- [Блокирование, разблокирование](#)
- [Обработка Ajax-запросов](#)

Логика основных действий для редактирования данных заложена в ядре системы. Это несколько файлов в папке *adm/includes*, которые описаны ниже. Они отвечают за базовые действия по управлению данными в административной части сайта. Это означает, что если [создать модуль](#) с простым содержимым, diafan.CMS автоматически подгрузит эти файлы, и данными можно будет управлять, используя основные действия.

В diafan.CMS предусмотрены следующие действия:

- **вывод данных в виде списка** – файл *adm/includes/show.php*;
- **редактирование, добавление данных** – файл *adm/includes/edit.php*;
- **валидация данных** – файл *adm/includes/validate.php*;
- **сохранение** – файл *adm/includes/save.php*;
- **удаление, удаление в корзину** – файл *adm/includes/del.php*;
- **публикация или скрытие данных** – файл *adm/includes/act.php*;
- **перемещение строки (сортировка и переопределение родителя)** – файл *adm/includes/move.php*;
- **обработка Ajax запросов** – файл *adm/includes/ajax.php*.

Внимание!

Этих действий обычно достаточно, чтобы управлять большинством наборов данных. Например, нужно создать модуль по управлению персоналом. Это будет таблица сотрудников с фамилиями. Создавая модуль, достаточно сделать таблицу сотрудников в БД, и базовый функционал сразу автоматически позволит: выводить список сотрудников, добавлять новых, редактировать имеющихся, удалять, сортировать, скрывать выборочных.

В файле модуля можно использовать как стандартный функционал действий, так и расширить его, и даже полностью переопределить на свои действия.

Рассмотрим, как это можно сделать.

Вывод данных в виде списка

При открытии модуля вызывается функция *show()*, определенная в каждом классе администрирования модуля. В этой функции можно заложить свое представление страницы администрирования модуля, можно вызвать стандартную функцию, формирующую список элементов таблицы модуля:


```
$this->diafan->list_row();
```

Можно вывести ссылку «Добавить новый элемент»:

```
$this->diafan->addnew_init('Добавить новый элемент');
```

По умолчанию в качестве ссылки на редактирование используется значение поля *name*. Это можно изменить с помощью свойства [\\$text_for_base_link](#).

Если для формирования списка нужно ввести дополнительное условие в SQL-запросе к базе данных, то воспользуйтесь свойством [\\$where](#).

Настройки вывода списка закладываются в переменную [\\$config](#), определенную в файле модуля.

В список можно добавить свои колонки. Для этого определяется массив [\\$config_other_row](#).

Редактирование, добавление

Если в URL страницы передана переменная *edit* или *addnew*, то подключается функция *edit()* из файла *adm/includes/edit.php*. Функция генерирует форму редактирования на основе массива [variables](#), определенного в классе модуля.

Функцию можно переопределить в классе модуля.

Также можно дополнить редактирование полей своим представлением вывода. Для этого в классе модуля определяется функция [edit_variable_переменная\(\)](#), или [edit_config_variable_переменная\(\)](#) для файла настроек.

Пользовательская функция вызывается всегда, если она определена, независимо от типа переменной. Исключение составляет тип *module*.

Некоторые пользовательские функции редактирования полей определены в файле *adm/includes/edit_functions.php*. Например, редактирование ЧПУ, сортировки, раздела сайта и пр. Их также можно переопределить в классе модуля.

Некоторые модули агрегируют информацию об элементах других модулей, и при их редактировании нужно чтобы связанная информация также редактировалась.

Например, при редактировании новости нужно редактировать прикрепленные изображения и теги.

Для этого в модулях-подключениях нужно создать файл *modules/модуль/admin/модуль.admin.inc.php*. В файле должен быть определен класс *Модуль_admin_inc* с функцией *edit()* или *edit_config()* для файла настроек.

Затем в основном модуле (в примере это модуль *новости*) в таблице [\\$variables](#) нужно указать переменную, названную так же, как модуль-подключение, и задать ей тип *module*.

Пример:

```
public $variables = array(
    'main' => array(
        'images' => 'module',
        ...
    ),
    ...
);
```

Валидация

Перед сохранением данные проверяются, и в случае ошибки выводится сообщение. За валидацию отвечает файл `adm/includes/validate.php`. diafan.CMS проверяет данные на основе массива `variables`, определенного в классе модуля.

Можно добавить свою валидацию полей. Для этого в классе модуля определяется функция `validate_variable_переменная()` или `validate_config_variable_переменная()` для файла настроек.

Пользовательская функция вызывается всегда, если она определена, независимо от типа переменной. Исключение составляет тип `module`.

Некоторые пользовательские функции редактирования полей определены в файле `adm/includes/validate_functions.php`. Например, период действия, дополнительные поля и пр. Их также можно переопределить в классе модуля.

Сохранение

Если в URL страницы передана переменная `save` или `save_new`, то подключается функция `save()` из файла `adm/includes/save.php`. Функция сохраняет данные из формы на основе массива `$variables`, определенного в классе модуля.

Функцию можно переопределить в классе модуля.

Также можно дополнить сохранение полей своей обработкой. Для этого в классе модуля определяется функция `save_variable_переменная()` или `save_config_variable_переменная()` для файла настроек.

Функция определяет:

- `$this->diafan->set_query("название_поля='%f'");`

Для определения маски обратитесь к [разделу «Базы данных»](#).

- `$this->diafan->set_value("значение_поля");`

Пользовательская функция вызывается всегда, если она определена, независимо от типа переменной. Исключение составляет тип `module`.

Некоторые пользовательские функции редактирования полей определены в файле `adm/includes/save_functions.php`. Например, время редактирования, сортировка, ЧПУ и пр. Их

также можно переопределить в классе модуля.

Некоторые модули агрегируют информацию об элементах других модулей, и при их сохранении нужно чтобы связанная информация также сохранялась.

Например, при сохранении новости нужно сохранять информацию о ссылках в меню на новость.

Для этого в модулях-подключениях нужно создать файл `modules/модуль/admin/модуль.admin.inc.php`. В файле должен быть определен класс `Модуль_admin_inc` с функцией `save()` или `save_config()` для файла настроек.

Затем в основном модуле (в примере это модуль *новости*) в таблице `$variables` нужно указать переменную, названную так же, как модуль-подключение, и задать ей тип `module`.

Пример:

```
public $variables = array(
    'main' => array(
        'menu' => 'module',
        ...
    ),
    ...
);
```

Удаление, удаление в корзину

При удалении элемента или нескольких элементов модуля подключается функция `del()` из файла `adm/includes/del.php`.

Эту функцию можно переопределить в классе модуля.

Кроме того, если в классе модуля определена функция `delete($del_id, $trash_id)`, то помимо заложенных в ядре действий она выполняется для каждого удаляемого элемента.

Надо учитывать, что если элементы удаляются в корзину, то сопутствующие им элементы тоже должны быть удалены в корзину. Поэтому вместо прямого SQL запроса на удаление лучше использовать функцию `del_or_trash_where`, определенную в файле `adm/includes/del.php`.

Пример:

```
/* При удалении товара нужно удалять его цены. Для этого в классе Shop_admin файла
modules/shop/admin/shop.admin.php определяем функцию: */
public function delete($del_id, $trash_id)
{
    $this->del_or_trash_where("shop_price", "good_id=".$del_id, $trash_id);
}
```

Некоторые модули агрегируют информацию об элементах других модулей, и при их удалении нужно чтобы связанная информация также удалялась. Например, при удалении

товара нужно удалять ссылку в меню на него и комментарии о товаре.

Для этого в модулях-подключениях нужно создать файл *modules/модуль/admin/модуль.admin.inc.php*. В файле должен быть определен класс *Модуль_admin_inc* с функцией *delete*.

string delete (string \$module_name, integer \$element_id, [integer \$trash_id = 0], [boolean \$is_category = false]) – удаляет элементы модуля, связанные с другими элементами.

- access: protected
- string \$module_name: название модуля, элемент которого удаляется
- integer \$element_id: номер удаляемого элемента
- integer \$trash_id: номер записи в корзине, с которой связано удаление
- integer \$is_category: удаляемый элемент – категория

Блокирование, разблокирование

При удалении элемента или нескольких элементов модуля подключается функция *act()* из файла *adm/includes/act.php*.

Эту функцию можно переопределить в классе модуля.

Некоторые модули агрегируют информацию об элементах других модулей, и при их блокировании нужно чтобы связанная информация также блокировалась.

Пример:

```
/* modules/tags/admin/tags.admin.inc.php  
Переопределение блокировки/разблокировки для модуля «Теги»  
*/  
public function act($table, $element_ids, $act)  
{  
    if(! DB::query_result("SELECT id FROM {modules} WHERE name='tags' LIMIT 1"))  
    {  
        return;  
    }  
    if (! $this->diafan->configmodules("tags"))  
    {  
        return;  
    }  
    if ($this->diafan->config('category'))  
    {  
        $table = str_replace('_category', '', $table);  
        $result = DB::query("SELECT id FROM {". $table ."} WHERE cat_id IN (%h)",  
            implode(',', $element_ids));  
        $element_ids = array();  
        while ($row = DB::fetch_array($result))  
        {  
            $element_ids[] = $row["id"];  
        }  
    }  
}
```

```

        if($element_ids)
        {
            DB::query("UPDATE {tags} SET [act]='%d' WHERE module_name='%h' AND
element_id IN (%h)", $act, $table, implode(',', $element_ids));
        }
    }
    else
    {
        DB::query("UPDATE {tags} SET [act]='%d' WHERE module_name='%h' AND element_id
IN (%h)", $act, $table, implode(',', $element_ids));
    }
}

```

Для этого в модулях-подключениях нужно создать файл *modules/модуль/admin/модуль.admin.inc.php*. В файле должен быть определен класс *Модуль_admin_inc* с функцией *act*.

string act (string \$table, array \$element_ids, integer \$act) – блокирует / разблокирует элементы модуля, связанные с другими элементами.

- access: protected
- string \$table таблица
- array \$element_ids: номера блокируемого / разблокируемого элементов
- integer \$act: блокировать/разблокировать

Обработка Ajax-запросов

Некоторые запросы требуют специальной обработки. Например, подгрузка списка дерева сайта при редактировании родителя страницы. Эта обработка происходит в файле *adm/includes/ajax.php*.

Если нужно производить обработку Ajax-запросов в модуле, то создается файл *modules/модуль/admin/модуль.admin.ajax.php*. В файле описывается класс *Модуль_admin_ajax*. Обработка данных производится в функции *ajax()*. Для того чтобы система подключила файл, нужно отправить переменную `$_POST["module"] = 'модуль';`

Администрирование модуля

Файл администрирования модуля – это файл `modules/модуль/admin/имя_модуля.admin.php`, в котором определен класс `Имя_модуля_admin`. Этот класс построен на основе единого каркаса, поэтому должен наследовать класс `Frame_admin`.

Если модуль имеет несколько частей, то административная часть модуля может содержать и другие файлы `modules/модуль/admin/имя_модуля.admin.часть.php`, где описан класс `имя_модуля_admin_часть`.

Свойства класса могут определять следующие переменные-настройки:

1. Название таблицы:

```
public $table = 'имя_таблицы_в_БД';
```

2. Описание полей:

```
public $variables = array (
    'название_группы_полей_1_латинскими_буквами' => array (
        'название_поля_1_латинскими_буквами' => array(
            'type' => 'тип_поля',
            'name' => 'Название_поля_русскими_буквами',
            'help' => 'Подсказка',
            // поле переводится на другие языки
            'multilang' => true,
            // значение по умолчанию
            'default' => 'значение',
            // поле нельзя отредактировать
            'disabled' => true,
            // не сохранять значение
            'no_save' => true,
            // высота поля визуального редактора
            'height' => 400,
        ),
        'название_поля_2_латинскими_буквами' => array(
            ...
        ),
        // сокращенная форма
        'название_поля_3_латинскими_буквами' => 'тип_поля',
        ...
    ),
    'название_группы_полей_2_латинскими_буквами' => array (
        ...
    ),
    ...
    // группа дополнительных полей
    'other_rows' => array (
        ...
    )
);
```

),
);

Определены следующие типы:

- text – строка;
- password – пароль;
- numtext – число;
- floattext – число с плавающей точкой;
- checkbox – переменная с двумя значениями (1,0);
- select – список (для использования типа необходимо определить массив `public $select["переменная"]` или `public $select_arr["переменная"]` см.ниже);
- date – дата (значение выводится в формате dd.mm.yyyy, сохраняется в [формате UNIX](#));
- datetime – дата и время (значение выводится в формате dd.mm.yyyy hh:mm, сохраняется в [формате UNIX](#));
- textarea – текстовое поле типа textarea;
- editor – текстовое поле с визуальным редактором;
- none – при редактировании не выводится, но значение сохраняется (можно использовать с пользовательской функцией);
- module – обозначаются поля, к которым подключены другие модули;
- function – обозначаются поля, к которым подключены пользовательские функции (подключаются также к любому другому типу кроме module);
- hr – выводит горизонтальную полосу на странице редактирования, при сохранении не учитывается.

Для удобного доступа к массиву `$variables` определены следующие функции:

- **variable** ([string \$key = ''], [string \$type_info = 'type'], [mixed \$value = NULL]) – возвращает, назначает информацию о поле.
 - string \$key – название переменной. Если не задано, то используется переменная текущей итерации цикла (при сохранении и редактировании);
 - string \$type_info – тип информации (type, name, multilang, disabled, default и др.);
 - mixed \$value – значение. Если задано, то значение переопределяется.

Пример:

```
if($this->diafan->variable('name') == 'editor')
{
    echo 'Поле name редактируется с помощью визуального редактора.';
}
if($this->diafan->variable('type', 'multilang'))
{
    echo 'Значение поля type переводится на другие языки.';
}
// запрещаем редактирование значения поля role_id
$this->diafan->variable('role_id', 'disabled', true);
```

- **is_variable** (string \$key) – возвращает, определена ли переменная в списке полей.
 - string \$key – название переменной.

Пример:

```
if($this->diafan->is_variable('act'))
{
    echo 'Поле act определено в списке полей.';
}
```

- **variable_unset (string \$key)** – удаляет переменную из списка полей.

- string \$key – название переменной.

Пример:

```
// удаляем поле site_id из списка полей
$this->diafan->variable_unset('site_id');
```

- **variable_name ([string \$key = "])** – возвращает название поля.
 - string \$key – название переменной. Если не задано, то используется переменная текущей итерации цикла (при сохранении и редактировании).

Пример:

```
echo 'Название текущего поля: '.$this->diafan->variable_name();
echo 'Название поля role_id: '.$this->diafan->variable_name('role_id');
```

- **variable_multilang ([string \$key = "])** – определяет, является ли поле мультиязычным.
 - string \$key – название переменной. Если не задано, то используется переменная текущей итерации цикла (при сохранении и редактировании).

Пример:

```
if($this->diafan->variable_multilang('name'))
{
    echo 'Значение поля name переводится на другие языки.';
}
```

- **variable_disabled ([string \$key = "], [boolean \$value = NULL])** – возвращает, назначает атрибут disabled для переменной.
 - string \$key – название переменной. Если не задано, то используется переменная текущей итерации цикла (при сохранении и редактировании);
 - boolean \$value – значение. Если задано, то значение переопределяется.

Пример:

```
if($this->diafan->variable_disabled('counter'))
{
    echo 'Поле counter нельзя отредактировать.';
}
// запрещаем редактировать поле created
$this->diafan->variable_disabled('created', true);
```


3. Справочники:

Если в массиве `public $variables` определены переменные с типом `select`, то для них необходимо определить массив `public $select["переменная"]` или `public $select_arr["переменная"]`.

Формат:

1. Использование значений из таблицы:

```
public $select["название_переменной"] = array(
    "название таблицы",
    "название поля значения",
    "название поля для отображения",
    "значение списка при добавлении элемента",
    "Отображение для пустого поля (если значение не задано, пустого элемента в
списке не существует",
    "условие для SQL-запроса",
    "поле для организации вложенности (если вложенности не существует, значение
пустое)"
);
```

2. Массив содержит значения для списка:

```
public $select_arr = array(
    "название_переменной" => array(
        "значение" => "отображение",
        ...
    )
);
```

Массив значений доступен через функцию `$this->diafan->select_arr("название")`;

Если функции передать второй аргумент, то она возвратит значение для ключа в массиве.
Формат:

```
$this->diafan->select_arr("название", "ключ_массива");
```

Этой функцией можно задать новое значение. Формат:

```
$this->diafan->select_arr("название", "ключ_массива", "новое значение");
```

4. Настройки отображения:

`public $config` – настройка отображения списка значений:

- `act` – отображение кнопки быстрой публикации (при наличии поля `act`);
- `category` – использование категорий в модуле (отдельная таблица {таблица_category}); часть с категориями;
- `category_flat` – категории элементов модуля не содержат вложенностей;
- `category_no_empty` – категория всегда выбрана;

- `category_no_multilang` – имя категории не переводится;
- `category_rel` – работают вместе с таблицей {модуль_category_rel};
- `config` – файл настроек модуля;
- `del` – отображение кнопки удаления;
- `date` – отображение даты в формате dd.mm.yyyy (при наличии поля data), сортировка по дате;
- `datetime` – отображение даты и времени в формате dd.mm.yyyy hh:ii (при наличии поля data), сортировка по дате;
- `element` – использование категорий в модуле (отдельная таблица {таблица_category}); часть с элементами;
- `element_multiple` – элемент может быть прикреплен к нескольким категориям;
- `element_site` – использование разделов (если модуль может быть подключен к нескольким страницам);
- `image` – показывать изображение элемента;
- `link_to_element` – основная ссылка ведет к списку элементов, принадлежащих категории;
- `menu` – показывать страницу в меню;
- `multiupload` – мультизагрузка изображений (подключение JS-библиотек);
- `no_out` – не отображать кнопку «Сохранить и выйти»;
- `porage` – показывать все элементы (без разбивки на страницы);
- `only_self` – показывать только материалы редактора, если это задано в правах пользователя;
- `order` – возможность сортировки элементов (при наличии поля sort);
- `parent` – использовать вложенность элементов (при наличии поля parent_id);
- `prior` – приоритетная сортировка по полю prior;
- `search_name` – форма поиска по названию;
- `tab_card` – использование вкладок;
- `trash` – использование корзины для удаления элементов;
- `view` – просмотр страницы на сайте.

Формат:

```
public $config = array(
    'переменная',
    ...
);
```

Настройки доступны через функцию `$this->diafan->config("название")`;

Этой функцией можно выключить или отключить настройку отображения. Формат:

```
$this->diafan->config("название", true|false);
```

5. Настройки для формирования списка элементов:

`public $where` – дополнительное условие для SQL-запроса к базе данных.

Пример:

В административной части список комментариев должен формироваться только из неактивных комментариев.

```
public $where = " AND act='0'";
```

`public $text_for_base_link` – текст для ссылки на редактирование в списке. По умолчанию используется значение поля *name*.

Формат:

```
public $text_for_base_link = array(  
    ['variable' => 'Название_поля_в_базе_данных'],  
    ['text' => 'Текст']  
);
```

Если не задано поле и текст, то ссылка не выводится.

Пример:

В модуле «Корзина» не нужно выводить ссылку на редактирование. В файле *modules/trash/admin/trash.admin.php*:

```
public $text_for_base_link = array();
```

Если задано поле, то для ссылки используется значение этого поля. Мультиязычность поля определяется по массиву `$variables`.

Пример:

В модуле «Комментарии» для ссылки нужно использовать текст комментария. В файле *modules/comments/admin/comments.admin.php*:

```
public $text_for_base_link = array(  
    'variable' => 'text'  
);
```

Если задан текст, то он используется для ссылки.

Пример:

В модуле «Обратная связь» в основной таблице нет информации, которую можно использовать для ссылки, поэтому выводим «Редактировать». В файле *modules/feedback/admin/feedback.admin.php*:

```
public $text_for_base_link = array(  
    'text' => 'Редактировать'  
);
```

Можно в текст подставить значение поля. Для этого в тексте указываем [описатель типа](#).

Пример:

В модуле «Магазин – Заказы» для ссылки нужно выводить «Заказ № 1, 2, ...». В файле *modules/shop/admin/shop.admin.order.php*:

```
public $text_for_base_link = array(  
    'variable' => 'id',  
    'text' => 'Заказ № %d'  
);
```

6. Дополнительные поля для формирования списка элементов:

`public $config_other_row` – массив названий дополнительных полей для отображения.

Формат:

```
public $config_other_row = array (  
    'название_поля_1_латинскими_буквами' => array(  
        'type' => 'тип_поля',  
        'class' => 'CSS-класс'  
    ),  
    // сокращенный вариант  
    'название_поля_2_латинскими_буквами' => 'тип_поля',  
    ...  
);
```

Определены следующие типы:

- function – вывод осуществляется пользовательской функцией;
- string – строка;
- select – значение списка из массива `$select_arr`;
- text – текст.

Для типа function необходимо определить пользовательскую функцию формата: `other_row_переменная($row)`, где `$row` – массив значений для текущего элемента. Пример использования модули *modules/rating/admin/rating.admin.php*, *modules/users/admin/users.admin.php*.

7. Прикрепление модуля к нескольким разделам сайта

Для возможности прикрепления модуля к нескольким страницам, необходимо вставить следующий код:

```
public $variables = array (  
    'любая_группа_полей' => array (  
        'site_id' => array(  

```

```
        'type' => 'function',  
        'name' => 'Раздел сайта',  
    ),  
);  
public $config = array('element_site', ...);
```

8. Значение поля зависит от значений другого поля

Если отображение одного поля зависит от значения другого, то используется массив `public $show_tr_click_checkbox`.

Формат:

```
public $show_tr_click_checkbox = array(  
    "название_главной_переменной" => array(  
        'название зависимой переменной',  
        ...  
    ),  
    ...  
);
```

Главная переменная должна иметь тип `checkbox`.

Пример:

В настройках модуля «Вопрос-Ответ» поля «Количество вопросов в списке категорий» и «Показывать вопросы подкатегорий» выводятся, если отмечена опция «Использовать категории». В файле `modules/faq/admin/faq.admin.config.php`:

```
public $show_tr_click_checkbox = array(  
    'cat' => array(  
        'count_list',  
        'childrens_elements',  
    ),  
);
```

9. Значения свойств задаются динамически

Если значения вышеописанных свойств модуля зависят от каких-либо условий, то эти условия описываются в функции `prepare_config()`.

Пример:

Если в настройках модуля «Фотогалерея» отключена опция «Использовать альбомы», то настройки «использование категорий в модуле» и «элемент может быть прикреплен к

нескольким категориям» должны быть отключены.

```
public function prepare_config()
{
    if(! $this->diafan->configmodules("cat", "photo", $this->diafan->site))
    {
        $this->diafan->config("element", false);
        $this->diafan->config("element_multiple", false);
    }
}
```

Настройки модуля

Настройки модуля – это интерфейс для конфигурирования модуля.

Настройки всех модулей хранятся в таблице `{config}`.

Для использования настроек модуля при формировании страницы как в пользовательской части, так и в административной, существует функция [configmodules](#).

Пример:

```
if($this->diafan->configmodules("переменная", "модуль") == "значение")
{
    //выполнение действия
}
```

Файл настроек модуля – `modules/модуль/admin/модуль.admin.config.php`. Файл строится по тому же принципу, что и любой файл административной части модуля.

Для того чтобы система поняла, что этот файл – настройки модуля, нужно указать:

```
public $config = array (
    'config',
    ...
);
```

Разделы сайта

Если модуль прикреплен к нескольким страницам сайта, то для каждой страницы можно задать свои настройки для модуля.

Это обеспечивает следующая запись в классе настроек модуля:

```
public $config = array (
    'element_site',
    ...
);
```

Подключение модуля

Для того чтобы модуль можно было редактировать, необходимо создать страницу в административной части, ссылающуюся на него.

Для этого существует специальный интерфейс, доступный по адресу:
<http://site.ru/admin/adminsite/>.

Внимание!

Важно знать, что для подключения модуля нужно, чтобы псевдоссылка страницы совпадала с названием модуля. Если подключается одна из неосновных частей модуля (файл *modules/модуль/admin/модуль.admin.часть.php*), то псевдоссылка должна быть *модуль/часть*.

Пример:

*modules/**clauses**/admin/**clauses**.admin.**category**.php* →
[http://example.com/admin/**clauses**/**category**/](http://example.com/admin/clauses/category/)

Новые страницы заносятся в таблицы *{adminsite}* и, если это страница второго уровня, *{adminsite_parents}*.

Подключения (API)

Подключения или API – набор готовых функций и команд в среде diafan.CMS, служащих для достижения определенных целей: работа с файлами, работа с базой данных, работа с изображениями и т.д.

- [База данных](#)
- [Валидация данных](#)
- [Изображения](#)
- [Кастомизация](#)
- [Конфигурация сайта](#)
- [Кэширование](#)
- [Маршрутизатор](#)
- [Общие функции](#)
- [Подключение шаблонов](#)
- [Пользователи](#)
- [Почта](#)
- [Режим разработки](#)
- [Сессии](#)
- [Сжатие страниц](#)
- [Установка модулей](#)
- [Файлы](#)
- [SMS](#)

База данных

- [Функции для работы с БД](#)

Класс с функциями для работы с базой данных – *DB* – определен в файле *includes/database.php*.

Все функции имеют access: public, что означает, что вызывать их в diafan.CMS можно откуда угодно, из любого файла и модуля. Вызываются функции в формате: *DB::имя_функции()*

Пример:

Например, чтобы сделать запрос к таблице **diafan_users**, и получить оттуда все поля с пользователями одного типа, нужно сделать так:

```
$result = DB::query("SELECT fio FROM diafan_users WHERE role_id=%d", $role_id);
```

Указывая таблицу, можно не писать префикс diafan_users, а написать {users}.

```
$result = DB::query("SELECT fio FROM {users} WHERE role_id=%d", $role_id);
```

Обратите внимание, в примере мы не пишем **WHERE role_id=\$role_id**, а пишем **WHERE role_id=%d**. Это сделано для безопасности, чтобы можно было спокойно делать запросы с внешними переменными типа *\$_GET["role_id"]*.

Для фильтрации аргументов SQL-запроса вместо данных передаются маски, а данные передаются в качестве дополнительных аргументов для функций *DB::query()*, *DB::query_result()* и *DB::query_range()* в порядке, в котором они следуют в SQL-запросе. Существуют следующие маски:

- *%d* – число,
- *%s* – строка,
- *%h* – строка без HTML кода,
- *%%* – символ %,
- *%f* – число с плавающей точкой,
- *%b* – файлы

Название таблицы пишется без префикса в фигурных скобках.

Пример:

```
{attachments}
```

Переводимые переменные в SQL-запросе пишутся в квадратных скобках.

Пример:

```
[name]
```

Для проверки, правильно ли сформировался SQL-запрос, или для отладки, запрос можно вывести на экран. Для этого нужно перед запросом написать «**DEV**» и включить [режим разработки](#).

Пример:

```
$this->result["text"] = DB::query_result("DEV SELECT [text] FROM {shop_category} WHERE id=%d", $this->diafan->cat);
```

Функции для работы с БД

resource **connect(string \$url)** – пробует подключиться к базе данных.

- string \$url: доступ к базе данных

Пример:

```
DB::connect("mysql://user_sql:pass@localhost/diafan_base");
```

Подключаемся, например, к другой БД на хостинге.

Пример:

Подключимся к другой базе данных и выведем список пользователей.

```
public function show_other_users()
{
    //подключаемся к другой БД
    DB::connect("mysql://root:@localhost/otherdb");
    $users = DB::query("SELECT id, fio FROM otherdb_users");
    while($row = DB::fetch_array($users))
    {
        echo $row["fio"] . "<br>; //выводим список пользователей
    }
    //возвращаем подключение обратно
    DB::connect("mysql://root:@localhost/mydb");
}
```

array **fetch_array(\$result)** – извлекает результирующий ряд как массив.

- \$result

Предположим, что нам нужно вывести на странице товара возможные способы доставки. Для этого отредактируем файл *shop.view.id.php*, добавив в конец следующий блок.

Пример:

```
/* modules/shop/views/shop.view.id.php
Добавляем запрос и формируем вывод
```

```

*/
$delivery = DB::query("SELECT [name] FROM {shop_delivery} ORDER BY sort ASC");
echo 'Способы доставки:';
while ($row = DB::fetch_array($delivery))
{
    echo $row["name"] . <br>;
}

```

object **fetch_object(\$result)** – извлекает результирующий ряд как объект.

- \$result

Усложним предыдущий пример и вернем способ доставки с id = 2 в виде объекта.

Пример:

```

/* modules/shop/views/shop.view.id.php
Добавляем запрос и формируем вывод объекта
*/
$delivery = DB::fetch_object(DB::query("SELECT * FROM {shop_delivery} WHERE id=%d
LIMIT 1", 2));
echo 'Способы доставки:';
echo $delivery->name1 . ' : ' . $delivery->text1;

```

integer **last_id (string \$table)** – получает номер *id* для таблицы *\$table*.

- string \$table: название таблицы без префикса

Пример:

```
$id = DB::last_id("rewrite");
```

Присваиваем переменной значение *ID*, который добавится в таблицу *diafan_rewrite* следующим.

Пример:

```

/* modules/cart/cart.ajax.php
При добавлении товара в корзину, в БД заносится соответствующая запись. Для
возможности добавления информации к ней, $order_id будет хранить id этой записи.
*/
DB::query("INSERT INTO {shop_order} (user_id, created, status, status_id, lang_id) VALUES
(%d, %d, '0', %d, %d)",
$this->diafan->_user->id,
time(),
$status_id,
_LANG
);
$order_id = DB::last_id("shop_order");

```

```
$this->insert_values(array("id" => $order_id, "table" => "shop_order", "params" => $params));
```

integer **num_rows(\$result)** – получает количество рядов в результате.

- \$result

Пример:

```
$result = DB::query("SELECT id FROM {news} WHERE created>%d", time());  
echo DB::num_rows($result);
```

Выводим количество новостей старше текущей даты.

Данную функцию часто приходится использовать для проверки количества строк, вернувшихся из запроса. Если продолжить пример с выводом способов доставки на странице товара, то мы можем использовать *num_rows* для контроля за пустым результатом.

Пример:

```
/* modules/shop/views/shop.view.id.php  
Если варианты доставки отсутствуют, то выведем сообщение об этом.  
*/  
$delivery = DB::query("SELECT [name] FROM {shop_delivery} WHERE trash='0' AND [act]='1'  
ORDER BY sort ASC");  
echo 'Способы доставки:';  
if(DB::num_rows($delivery) > 0)  
{  
    while ($row = DB::fetch_array($delivery))  
    {  
        echo $row["name"] . <br>;  
    }  
}  
else  
{  
    echo 'В данный момент доставка не доступна.';  
}
```

resource **query(\$query)** – отправляет запрос к базе данных.

- \$query

Пример:

```
DB::query("INSERT INTO {clauses} ([name], [act], created) VALUES ('%h', '%d', %d)",  
$_POST["name"], $_POST["act"], time());
```

Вставляем в таблицу *diafan_clauses* данные.

Подобные запросы встречаются повсюду в CMS. Аналог обычного запроса SQL.

resource **query_range(\$query)** – отправляет запрос к базе данных с лимитом на количество получаемых в результате рядов.

- \$query

Пример:

```
$result = DB::query_range("SELECT name FROM {image} WHERE module_name='shop' AND element_id=%d", $good_id, 0, 3);`
```

Получаем имена трех картинок, прикрепленных к товару.

Подобные запросы вы можете часто встретить в файлах **.model.php*, где производится выборка данных для списков, а также в модулях, поддерживающих постраничную навигацию.

Пример:

```
/* modules/photo/photo.model.php  
Запрос для вывода списка фотографий.  
*/  
private function list_query($time)  
{  
    $result = DB::query_range(  
        "SELECT e.id, e.[name], e.[anons], e.timeedit, e.cat_id, e.site_id FROM {photo} AS e"  
        .($this->diafan->_user->id ? " LEFT JOIN {access} AS a ON a.element_id=e.id AND"  
        a.module_name='photo'" : "" )  
        . " WHERE e.[act]='1' AND e.trash='0' AND e.site_id=%d"  
        . " AND e.date_start<=%d AND (e.date_finish=0 OR e.date_finish>=%d)"  
        . " AND (e.access='0'"  
        .($this->diafan->_user->id ? " OR e.access='1' AND"  
        a.role_id=".$this->diafan->_user->role_id : "  
        .")"  
        . " GROUP BY e.id ORDER BY e.sort ASC, e.id ASC",  
        $this->diafan->cid, $time, $time,  
        $this->diafan->_paginator->polog, $this->diafan->_paginator->nastr  
    );  
    return $result;  
}
```

\$this->diafan->_paginator->polog, **\$this->diafan->_paginator->nastr** в данном случае выступают в роли начального и конечного индекса выборки.

mixed **query_result(\$query)** – получает результирующие данные из SQL-запроса.

- \$query

Пример:

```
$text = DB::query_result("SELECT [text] FROM {shop_category} WHERE id=%d", $this->
```

```
>diafan->cat);
```

Получаем описание текущей категории из БД.

Давайте вернемся к нашему примеру со способами доставки и сразу вернем интересующую нас строку.

Пример:

```
/* modules/shop/views/shop.view.id.php
```

Теперь мы можем обращаться к `$delivery` как к переменной, содержащей строку, а не как к массиву.

```
*/
```

```
$delivery = DB::query_result("SELECT [name] FROM {shop_delivery} WHERE id=%d AND  
trash='0' AND [act]='1'", 2);  
echo 'Способы доставки:';  
echo $delivery;
```

`mixed result($result, [ $row = 0])` – получает результирующие данные.

- `$result`
- `$row`

Этот метод используется внутри класса *database*

Пример:

```
/* includes/database.php
```

Пример использования в функции `last_id($table)` класса *database*

```
*/
```

```
public static function last_id($table)  
{  
    return self::result(self::query("SELECT max(id) FROM {". $table. "} limit 1"));  
}
```

`mixed title(string $table, integer $id, string $param_name)` – возвращает значение поля `$param_name` для таблицы `$table` и строки с номером `$id`.

- `string $table`: название таблицы без префикса
- `integer $id`: идентификатор строки
- `string $param_name`: название поля

Пример:

```
echo DB::title("users", 3, "fio");
```

Выводим ФИО пользователя с `id=3` из таблицы `{users}`.

Иногда, в случае простых запросов, это может упростить работу с БД

Пример:

/ modules/update/admin/update.admin.list.php*

В функции `other_row_user_id($row)` используется для вывода имени пользователя, добавившего файл

**/*

```
public function other_row_user_id($row)
{
    return "
        .($row["user_id"] ?
        .BASE_PATH_HREF.'users/edit'. $row["user_id"].';/'>'.DB::title("users", $row["user_id"], "fio").
        : "");
}
```

mixed where(string \$table, string \$where, string \$param_name) – возвращает значение поля `$param_name` для таблицы `$table` и одной строки, соответствующей запросу `$where`.

- string `$table`: название таблицы без префикса
- string `$where`: SQL-условие
- string `$param_name`: название поля

Пример:

```
DB::where("people", "parent_id=3 AND act=1", "name");
```

Выводим поля `name` из таблицы `people` с условием что они активны и родительская категория = 3.

Пример:

/ modules/shop/views/shop.view.id.php*

Выводим значение поля `name` для записи с `id = 2`

**/*

```
$delivery = DB::where("shop_delivery", "id=2 AND trash='0' AND [act]='1'", "[name]");
echo 'Способы доставки:';
echo $delivery;
```

string escape_string(string \$str) – Экранирует специальные символы в строке для использования в операторе SQL с учётом текущего набора символов/charset соединения.

- string `$str`: исходная строка

Пример:

/ modules/forum/forum.model.php*

Функция `list_search()` осуществляет поиск по заданному слову

**/*

```
$search = "";
if (isset($_GET["searchword"]))
{
```



```
if (is_array($_GET["searchword"]))
{
    $_GET["searchword"] = "";
}
$search = DB::escape_string(htmlspecialchars(stripslashes($_GET["searchword"])));
}
$this->result["value"] = $search;
```

Валидация данных

- [Методы](#)

Методы для валидации данных объединены в класс *Validate*, описанный в файле *includes/validate.php*.

Они могут использоваться в [моделях](#) и [обработчиках AJAX запросов](#) непосредственно для проверки переданных данных.

Методы

string|boolean **login(string \$name, [integer \$id = 0])** – проверяет логин на валидность.

- return: false
- access: public
- string \$name: логин
- integer \$id: номер пользователя

Пример:

```
$mes = Validate::login($_POST["name"]);  
if ($mes)  
    $this->result["errors"]["name"] = $this->diafan->_($mes, false);
```

string|boolean **mail(string \$mail)** – проверяет e-mail на валидность.

- return: false
- access: public
- string \$mail: электронный ящик

Пример:

```
$mes = Validate::mail($_POST["mail"]);  
if ($mes)  
    $this->result["errors"]["mail"] = $this->diafan->_($mes, false);
```

string|boolean **mail_user(string \$mail, [integer \$id = 0])** – проверяет, зарегистрирован ли пользователь с указанным электронным ящиком.

- return: false
- access: public
- string \$mail: электронный ящик
- integer \$id: номер пользователя

Пример:

```
$mes = Validate::mail_user($_POST["mail"]);  
if ($mes)  
    $this->result["errors"]["mail"] = $this->diafan->_($mes, false);
```

string|boolean **password(string \$password, [boolean \$is_simple = false])** – проверяет пароль на валидность. Если передана переменная `$is_simple = true`, то пароль сравнивается с базой простых паролей, описанных в файле `includes/validate.simple_passwords.php`.

- return: false
- access: public
- string \$password: пароль
- boolean \$is_simple: проверять сложность пароля

Пример:

```
$mes = Validate::password($_POST["password"]);  
if ($mes)  
    $this->result["errors"]["password"] = $this->diafan->_($mes, false);
```

string|boolean **numtext(string \$value)** – проверяет число.

- return: string|boolean
- access: public
- string \$value: значение

string|boolean **floattext(string \$value)** – проверяет число с плавающей точкой.

- return: string|boolean
- access: public
- string \$value: значение

string|boolean **date(string \$value)** – проверяет дату.

- return: string|boolean
- access: public
- string \$value: значение

string|boolean **datetime(string \$value)** – проверяет дату и время.

- return: string|boolean
- access: public
- string \$value: значение

string|boolean **text(string \$text, [integer \$max_lenght = 40])** – проверяет текст на наличие длинных слов.

- return: false
- access: public
- string \$text: текст
- integer \$max_lenght: максимальная длина слова

boolean **url(string \$url, [boolean \$absolute = false])** – проверяет URL на валидность.

- access: public
- string \$url: электронный адрес
- boolean \$absolute: абсолютный адрес

Изображения

- [Методы](#)

Для работы с изображениями в файле *includes/image.php* определен класс *Image*.

Методы

boolean `resize` (string `$src_image`, integer `$dest_width`, integer `$dest_height`, [integer `$quality = 80`], [boolean `$max = FALSE`]) – изменяет размеры изображения.

- access: public
- string `$src_image`: путь к файлу
- integer `$dest_width`: новая ширина изображения
- integer `$dest_height`: новая высота изображения
- integer `$quality`: качество изображения
- boolean `$max`: изменять по максимальной стороне

Данный метод широко используется каждый раз, когда вы загружаете новое графическое изображение, либо изменяете размерность уже имеющихся файлов. Главный «эксплуататор» данного метода – класс *Images_inc*.

Пример:

```
/* modules/images/images.inc.php
```

При добавлении изображения, например, при создании новости, происходит добавление миниатюры для административной панели. Вот пример из функции `upload`.

```
*/
```

```
copy(ABSOLUTE_PATH.USERFILES."/original/".$new_name,  
ABSOLUTE_PATH.USERFILES."/small/".$new_name);  
Image::resize(ABSOLUTE_PATH.USERFILES."/small/".$new_name, $this->small[0], $this->  
small[1], $this->small[2]);
```

boolean `crop` (string `$original`, integer `$width`, integer `$height`, integer `$quality`, string `$vertical`, integer `$sy`, string `$horizontal`, integer `$sx`) – обрезает изображение.

- access: public
- string `$original`: путь к файлу
- integer `$width`: новая ширина изображения
- integer `$height`: новая высота изображения
- integer `$quality`: качество изображения
- string `$vertical`: вертикальное расположение знака (top, middle, bottom)
- integer `$sy`: отступ по вертикале
- string `$horizontal`: горизонтальное расположение знака (left, center, right)
- integer `$sx`: отступ по горизонтале

Данный метод будет использоваться в том случае, если вы выбрали метод обрезки при добавлении нового изображения. Подробнее о настройках изображений вы можете

прочитать в [соответствующем разделе](#).

Пример:

```
/* modules/images/images.editor.php
```

После загрузки изображения, посредством функции `selectarea()` пользователю предлагается выделить область, после чего происходит ее обрезка.

```
*/
```

```
Customization::inc("includes/image.php");  
Image::crop($path, $_POST["x2"] - $_POST["x1"], $_POST["y2"] - $_POST["y1"],  
$variation["quality"], 'top', $_POST["y1"], 'left', $_POST["x1"]);  
$this->diafan->_images->get_variation_image($file_name, 'editor', $variation, false, true);
```

boolean **watermark** (string **\$original**, string **\$watermark**, integer **\$quality**, string **\$vertical**, integer **\$sy**, string **\$horizontal**, integer **\$sx**) – добавляет водяной знак на изображение.

- access: public
- string **\$original**: путь к файлу
- string **\$watermark**: путь к водяному знаку
- integer **\$quality**: качество изображения
- string **\$vertical**: вертикальное расположение знака (top, middle, bottom)
- integer **\$sy**: отступ по вертикали
- string **\$horizontal**: горизонтальное расположение знака (left, center, right)
- integer **\$sx**: отступ по горизонтали

Изображение, используемое в качестве водяного знака, по умолчанию расположено в директории `userfiles/watermark/`.

Пример:

```
/* modules/images/images.inc.php
```

Функция `get_variation_image`. Если в настройках выбрано использование водяного знака, то он будет применен ко всем изображениям этого типа.

```
*/
```

```
case 'watermark':  
    Image::watermark($path, ABSOLUTE_PATH.USERFILES."/watermark/".$_action["file"],  
$variation["quality"], $_action["vertical"], $_action["vertical_px"], $_action["horizontal"],  
$_action["horizontal_px"]);  
    break;
```

boolean **wb** (string **\$original**, integer **\$quality**) – обесцвечивает изображение.

- access: public
- string **\$original**: путь к файлу
- integer **\$quality** качество изображения

Пример:

```
/* modules/images/images.inc.php
```

Функция `get_variation_image`. Если в настройках выбрано обесцвечивание, то к

изображению будет применен фильтр IMG_FILTER_GRAYSCALE, переводящий его в оттенки серого.

*/

case 'wb':

Image::wb(\$path, \$variation["quality"]);

break;

Кастомизация

Класс для внедрения пользовательских разработок – *Customization* – определен в файле *includes/customization.php*.

Этот класс позволяет дорабатывать файлы diafan.CMS так, чтобы при автообновлении изменения не затерлись. Для этого все доработки нужно размещать в файле с таким же названием, но суффиксом *.custom*.

Пример:

Для файла *modules/news/news.model.php* кастомизированный файл *modules/news/news.model.custom.php*.

В кастомизированном файле можно переопределить классы, свойства и функции классов. Причем можно не только переопределить функцию, но и добавить код в начало или конец. Для этого в кастомизированном файле определяется класс с аналогичным оригиналу именем. В классе определяются свойства и функции.

Свойства могут иметь следующие приставки:

- *replace* – переопределение свойства;
- *new* – новое свойство.

Функции могут иметь следующие приставки:

- *replace* – переопределение функции;
- *new* – новая функция;
- *before* – добавить код в начало функции;
- *after* – добавить код в конец функции.

Пример:

Дорабатываем файл *includes/user.php*. Для этого создаем файл *includes/user.custom.php* следующего содержания:

```
class User extends Core
{
    // делаем свойство класса публичными и задаем ему значение по умолчанию
    replace public $user = NULL;

    // добавляем новое свойство класса
    new public $age;

    // добавляем код в начало функции set()
    before public function set($user)
    {
        $current_user_id = $user->id;
    }
}
```



```
// заменяем функцию logout()
replace public function logout()
{
    return true;
}

// добавляем код в конце функции log()
after private function _log()
{
    $_SESSION["user_log"] = true;
}

// добавляем новую функцию check
new private function check()
{
    $_SESSION["check"] = true;
}
}
```

Новый файл сгенерируется и запишется в кэш. Подключается файл из кэша с той же скоростью, что и обычный PHP-файл, поэтому кастомизация не отразится на скорости генерирования страницы. Файл генерируется каждый раз, если включен режим разработки в параметрах сайта.

Конфигурация сайта

- [Методы](#)

Работа с файлом *config.php* осуществляется с помощью подключения «Конфигурация сайта» (файл *includes/config.php*, класс *Config*).

Методы

boolean true **save (array \$new_values, array \$languages)** – данный метод является системным и используется для сохранения параметров сайта из административной панели.

- access: public
- array \$new_values: новые значения параметров сайта
- array \$languages: языки сайта

Пример:

```
//изменяем параметр «Режим разработки»
include_once(ABSOLUTE_PATH.'includes/config.php');

$new_values = array('MOD_DEVELOPER' => true);
$languages = array(
    array('id' => 1),
    array('id' => 2)
);
Config::save($new_values, $languages);
```

Кэширование

- [Методы](#)

Кэширование обеспечивает класс *Cache*, описанный в файле *includes/cache.php*.

У подключения есть два бэкенда: файловое кэширование и кэширование Memcached. По умолчанию подключается файловое кэширование. В параметрах сайта можно подключить кэширование Memcached.

Время актуальности кэша не ограничено. Кэш чистится при редактировании контента модуля из административной части.

В [модуле «Параметры сайта»](#) предусмотрены две опции для управления кэшем: «Отключить кэширование» и «Сбросить кэш».

Методы

boolean delete(string|array \$name, [string \$module = "]) – удаляет кэш для модуля *\$module* с меткой *\$name*. Если функция вызвана с пустой меткой, то удаляется весь кэш для модуля *\$module*.

- return: true
- access: public
- string|array \$name: метка кэша
- string \$module: название модуля

Пример:

```
/* adm/includes/act.php  
Используется в функции act, сбрасывает кэш при изменении активности элемента  
*/  
$this->diafan->_cache->delete("", $this->diafan->module);
```

mixed get(string|array \$name, string \$module, [boolean \$ignore_mod_dev_cache = FALSE])
– читает кэш модуля *\$module* с меткой *\$name*.

- access: public
- string|array \$name: метка кэша
- string \$module: название модуля
- boolean \$ignore_mod_dev_cache: игнорировать отключение кэша в параметрах сайта

Пример:

```
/* includes/function.php  
Запрашиваются данные из кэша. Если их нет, то производится новая выборка,  
иначе используются закэшированные результаты.  
*/
```

```

$cache_meta = array(
    "name" => "page",
    "id" => $this->diafan->cid,
    "lang_id" => _LANG
);

$page = $this->diafan->_cache->get($cache_meta, 'site');

```

boolean save(mixed \$data, string|array \$name, string \$module, [boolean \$ignore_mod_dev_cache = FALSE]) – сохраняет данные *\$data* для модуля *\$module* с меткой *\$name*.

- return: true
- access: public
- mixed \$data: данные, сохраняемые в кэше
- string|array \$name: метка кэша
- string \$module: название модуля
- boolean \$ignore_mod_dev_cache: игнорировать отключение кэша в параметрах сайта

Пример:

```

/* includes/function.php
Производится кэширование данных для повторного использования.
*/
//сохранение кэша
$this->diafan->_cache->save($page, $cache_meta, 'site');

```

Пример:

```

/*
Пример использования методов класса Cache
*/
$cache_meta = array("name" => "block", "cat_id" => $cat_id, "language" => _LANG, ...);

if(! $result = $this->diafan->_cache->get($cache_meta, "модуль"))
{
    // генерирование кэша
    // сохранение кэша
    $this->diafan->_cache->save($result, $cache_meta, "модуль");
}

```

Маршрутизатор

- [Методы](#)

Route – класс для работы с ЧПУ. Описан он файле *includes/route.php*. Объект класса создается в административной и пользовательской частях и в модулях доступен для обращения через переменную `$this->diafan->_route`. Класс используется в [моделях](#) модулей и [обработчиках AJAX запросов](#).

Методы

string link(integer \$site_id, [string \$module_name = "], [integer \$cat_id = 0], [integer \$element_id = 0], [integer \$param_id = 0], [boolean \$insert_route_end = true]) – генерирует ссылку.

- access: public
- integer \$site_id: номер страницы сайта
- string \$module_name: модуль
- integer \$cat_id: номер категории
- integer \$element_id: номер элемента модуля
- integer \$param_id: номер параметра
- boolean \$insert_route_end: добавлять окончание ЧПУ в конце ссылки

Пример:

```
/* modules/shop/shop.model.php  
SELECT id, site_id, cat_id, name FROM shop  
получение ссылки на товар */  
$row["link"] = $this->diafan->_route->link($row["site_id"], "shop", $row["cat_id"], $row["id"]);
```

string module(string \$module_name, [boolean \$route_end = false]) – получает ЧПУ страницы сайта по названию модуля.

- access: public
- string \$module_name: название модуля
- boolean \$route_end: выводить окончание

Пример:

```
/* modules/cart/cart.model.php  
получение ссылки на страницу корзины */  
$result["link"] = BASE_PATH_HREF.$this->diafan->_route->module("cart", true);
```

integer|boolean false id_module(string \$module_name, mixed \$site_id = 0, boolean \$return_array = true) – определяет номер страницы, к которой прикреплен модуль, доступный текущему пользователю.

- access: public

- string \$module_name: название модуля
- mixed \$site_id: номер страницы (если задан, определяет, прикреплен ли модуль, есть ли доступ)
- boolean \$return_array: вернуть массив (или один номер)

Пример:

```
/* modules/registration/registration.model.php
определение существования модуля messages */
$message_site_id = $this->diafan->_route->id_module("messages", 0, false);
```

string **current_link**([string|array \$exclude = ''], [array \$include = array()]) – выдает URL текущей страницы с включенными или исключенными переменными.

- access: public
- string|array \$exclude: исключенные переменные
- array \$include: включенные переменные

Пример:

```
/* modules/messages/messages.ajax.php
определение текущей страницы, без параметра /page/ */
$this->result['redirect'] = BASE_PATH_HREF.$this->diafan->_route->current_link("page");
```

string **current_admin_link**([string|array \$exclude = '']) – выдает URL текущей страницы административной части с включенными переменными.

- access: public
- string|array \$exclude: исключенные переменные

Пример:

```
/* modules/forum/admin/forum.admin.category.php
определение текущей страницы из административной части модуля */
$link .= $this->diafan->_route->current_admin_link(array ( "page", "parent" ))
. 'cat' . $row["id"] . '/';
```

string **replace_link_to_id**(string \$text) – заменяет все ссылки в тексте на идентификаторы, чтобы ссылки были не зависимы от изменения ЧПУ.

- access: public
- string \$text: исходный текст

Пример:

```
/* adm/includes/save.php
при сохранении заменяем ссылки в тексте на идентификаторы */
$text = $this->diafan->_route->replace_link_to_id($text);
```

string **replace_id_to_link**(string \$text) – заменяет идентификаторы ссылки на ЧПУ.

- access: public
- string \$text: исходный текст

Пример:

```
/* includes/function.php
замена всех идентификаторов в тексте на ссылки */
$text = $this->diafan->_route->replace_id_to_link($this->diafan->text);
```

array|boolean false **search**(string \$rewrite, [boolean \$arguments_in_url = false]) – ищет псевдоссылку в базе данных.

- access: public
- string \$rewrite: текущая псевдоссылка
- boolean \$arguments_in_url: в URL переданы аргументы

Пример:

```
print_r( $this->diafan->_route->search('shop') );
/* вернет Array
(
  [site_id] => 11
  [module_name] => site
  [cat_id] => 0
  [element_id] => 0
  [param_id] => 0
  [rewrite] => shop
) */
```

boolean true **get_rewrite**(string \$module_name, integer \$id, [string \$tag = ""]) – получает ЧПУ по тегу.

- access: private
- string \$module_name: модуль
- integer \$id: номер элемента
- string \$tag: тег

Общие функции

Общие функции принадлежат к абстрактному классу *Core*, который наследуются другими классами. Эти функции доступны через основной объект `$diafan`, создание которого иницирует генерацию страницы. В модулях функции доступны для вызова через объект `$this->diafan`.

Пример:

```
/*  
Пример использования $this->diafan.  
*/  
$this->diafan->configmodules("list_img");
```

`void redirect([string $url = "], [integer $http_response_code = 302])` – редирект. Адрес для редиректа можно передавать как в абсолютном виде (*http://www.site.ru/*), так и в относительном (*news/*).

- access: public
- string \$url: URL для редиректа
- integer \$http_response_code: статус-код

Пример:

```
/* modules/config/admin/config.admin.php  
При успешном сохранении параметров сайта в административной панели происходит  
перевод на страницу success1/  
*/  
if ($admin_folder == ADMIN_FOLDER)  
{  
    $this->diafan->redirect(URL.'success1/');  
}  
else  
{  
    $this->diafan->redirect('http://'.BASE_URL.'/'.$admin_folder.'/config/success1/');  
}
```

`void redirect_js([string $url = "])` – редирект с помощью JavaScript. Используется в случае, если нельзя использовать редирект с помощью заголовков страниц.

- access: public
- string \$url: URL для редиректа

Пример:

```
/* modules/shop/admin/shop.admin.import.php  
В функции remove() после удаления импортированных элементов с сайта вызывается  
перенаправление на страницу error3/.  
*/
```



```
$this->diafan->redirect_js(URL.'error3/');
```

mixed **get_param**(array \$array, string \$name, [mixed \$default = null], [integer \$mask = 0]) – функция получает из массива переменную и приводит ее к типу в соответствии с маской (0 – вырезает все пробелы, 1 – вырезает тэги, 2 – оставляет только числа).

- access: public
- array \$array: массив с переменной
- string \$name: имя переменной в массиве
- mixed \$default: значение по-умолчанию
- integer \$mask: тип преобразования

Пример:

```
/* modules/registration/registartion.ajax.php
```

```
В функции send_mails() формируется тело сообщения, при этом поля заполняются данными, взятыми из запроса.
```

```
*/
```

```
$message = str_replace(  
array('%login', '%title', '%url', '%fio', '%email', '%params'),  
array(  
    $this->diafan->get_param($_POST, "name", "", 1),  
    TITLE,  
    BASE_URL,  
    $this->diafan->get_param($_POST, "fio", "", 1),  
    $this->diafan->get_param($_POST, "mail", "", 1),  
    $this->message_admin_param,  
),  
$this->diafan->configmodules('message_admin', "users")  
);
```

mixed **configmodules** (string \$name, [string \$module_name = 'site'], [integer \$site_id = 0], [integer \$lang_id = 0], [boolean \$value = false]) – функция возвращает значение переменной \$name в конфигурации модуля \$module_name для языковой версии \$lang_id и страницы \$site_id. Если задано значение \$value, функция записывает новое значение.

- access: public
- string \$name: имя переменной в конфигурации
- string \$module_name: название модуля
- integer \$site_id: раздел сайта
- integer \$lang_id: номер языковой версии
- boolean \$value: новое значение

Пример:

```
/* modules/registration/registartion.ajax.php
```

```
В функции fast_validate() при проверке введенных данных учитывается, включено ли использование электронной почты в качестве логина.
```

```
*/
```

```
if(! $this->diafan->configmodules("mail_as_login", "users") && $mes =
```

```

Validate::login($_POST["value"]))
{
    echo $this->diafan->_($mes, false);
}

```

string **short_text (string \$text, [integer \$length = 80])** – сокращает текст.

- access: public
- string \$text: исходный текст
- integer \$length: количество символов для сокращения

Пример:

```

/* modules/faq/faq.model.php
В функции get_elements($result) используется для формирования вывода вопроса и
ответа на него.
*/
$row["anons"] = $this->diafan->short_text($row["anons"], 160);
$row["text"] = ! empty($row["text"]) ? $this->diafan->short_text($row["text"], 160) : "";

```

string **prepare_xml (string \$text)** – подготавливает текст для отображения в XML-файле.

- access: public
- string \$text: исходный текст

Пример:

```

/* modules/news/news.rss.php
Используется для формирования xml, например при выводе rss-ленты новостей.
*/
$items .= "

.BASE_PATH_HREF.$link."
".$this->diafan->prepare_xml($row['anons'])."
.date("D, d F Y H:i:s T", $row['created']).""
.($this->diafan->configmodules("comments", "news", $row["site_id"]) ? "
.BASE_PATH_HREF.$link."" : "")."
";

```

string **convert(integer \$size)** – конвертирует количество бит в байты, килобайты, мегабайты.

- access: public
- integer \$size: размер в байтах

string **search_theme(intger \$id)** – выводит название шаблона для страницы с номером id.

- access: public
- integer \$size: размер в байтах

Пример:

```
/* modules/attachments/attachments.inc.php
При отправке прикрепленного файла уточняется его размер.
*/
$row["size"] = $this->diafan->convert($row["size"]);
```

array **get_children**(integer \$id, string \$table, [boolean \$trash = true]) – выдает массив номеров «детей».

- access: public
- integer \$id: номер исходного элемента
- string \$table: таблица
- boolean \$trash: не учитывать элементы, удаленные в корзину

Пример:

```
/* modules/menu/admin/menu.admin.inc.php
При удалении родительского пункта меню происходит удаление всех дочерних.
*/
$children = $this->diafan->get_children($id, "menu", true);
$children[] = $id;
DB::query("DELETE FROM {menu_parents} WHERE parent_id IN (%s)", implode(',',
$children));
DB::query("DELETE FROM {menu_parents} WHERE element_id IN (%s)", implode(',',
$children));
DB::query("DELETE FROM {menu} WHERE id IN (%s)", implode(',', $children));
```

array **get_parents**(integer \$id, string \$table) – выдает массив номеров родителей.

- access: public
- integer \$id: номер исходного элемента
- string \$table: таблица

Пример:

```
/* includes/function.php
Пример использования в функции show_path($attributes) для получения всех
родительских страниц для данной.
*/
$parents = $this->diafan->get_parents($this->diafan->cid, 'site');
$rparents = array();
$result = DB::query("SELECT id, [name], parent_id FROM {site} WHERE id IN (%h)",
implode(',', $parents));
```

string **encrypt**(string \$value) – кодирует пароль.

- access: public
- string \$value: исходный пароль

Пример:

```
/* includes/user.php
```

Пример использования при определении правильности пароля при входе в систему.

```
*/
```

```
$result = DB::query("SELECT * FROM {users} u WHERE trash='0' AND act='1' AND LOWER(".  
$name.")=LOWER('%s') AND password='%s'", trim($form_values['name']),  
encrypt(trim($form_values['pass'])));
```

string **translit(string \$text)** – переводит кириллицу в транслит для строки text.

- access: public
- string \$text: исходный текст

Пример:

```
/* modules/images/admin/images.admin.inc.php
```

При сохранении изображений их имена преобразуются в транслит.

```
*/
```

```
foreach($_POST["images_variation_name"] as $i => $name)  
{  
    $variations[] = array("name" => $this->diafan->translit($name), "id" =>  
$_POST["images_variation_id"][$i]);  
    $new_variations[] = $_POST["images_variation_id"][$i];  
}
```

integer **unixdate(string \$date)** – возвращает дату, переданную в формате dd.mm.yyyy hh:ii в виде даты в [формате UNIX](#).

- access: public
- string \$date: дата в формате dd.mm.yyyy hh:ii

Пример:

```
/* modules/shop/shop.1c.php
```

Пример использования в запросе при импорте 1С.

```
*/
```

```
$result = DB::query("SELECT * FROM {shop_order} WHERE created>%d AND trash='0'",  
$this->diafan->unixdate(LAST_1C_EXPORT));
```

string **formate_from_date(integer \$date)** – переводит дату из в формата гggг-мм-дд в формат дд.мм.гггг.

- access: public
- string date: дата в формате гggг-мм-дд

Пример:

```
/* modules/comments/comments.inc.php
```

Преобразование может использоваться в поле типа «Дата» в модуле комментариев.

```
*/
```

```
case "date":
```

```

if (! empty($values[$row["id"]][0]["value"]))
{
    $param[] = array(
        "id" => $row["id"],
        "name" => $row["name"],
        "value" => $this->diafan->formate_from_date($values[$row["id"]][0]["value"]),
        "value_id" => $values[$row["id"]][0]["id"],
        "type" => $row["type"]
    );
}
break;

```

string **formate_from_datetime(integer \$date)** – переводит дату из формата гggг-мм-дд чч:мм в формат дд.мм.гггг чч:мм.

- access: public
- string date: дата в формате гggг-мм-дд чч:мм

Пример:

```

/* modules/shop/shop.model.php
Преобразует данные в полях типа datetime.
*/
case "datetime":
    if ( ! empty($values[$row["id"]][0]["rvalue"]))
    {
        $param[] = array(
            "id" => $row["id"],
            "name" => $row["name"],
            "value" => $this->diafan->formate_from_datetime($values[$row["id"]][0]["rvalue"]),
            "value_id" => $values[$row["id"]][0]["id"],
            "type" => $row["type"]
        );
    }
    break;

```

string **formate_in_date(integer \$date)** – переводит дату из в формата дд.мм.гггг в формат гggг-мм-дд.

- access: public
- string date: дата в формате дд.мм.гггг

Пример:

```

/* modules/shop/admin/shop.admin.import.php
Используется при обратном преобразовании поля типа date.
*/
case 'date':
    if($error = Validate::date($value))
    {
        $this->error_validate('param', $error, $param["name"]);
    }

```

```

        $value = "";
    }
    else
    {
        $value = $this->diafan->formate_in_date($value);
    }
    break;

```

string **formate_in_datetime(integer \$date)** – переводит дату из в формата дд.мм.гггг чч:мм в формат гггг-мм-дд чч:мм.

- access: public
- string date: дата в формате дд.мм.гггг чч:мм

Пример:

```

/* modules/shop/admin/shop.admin.import.php
Используется при обратном преобразовании поля типа datetime.
*/
case 'datetime':
    if($error = Validate::datetime($value))
    {
        $this->error_validate('param', $error, $param["name"]);
        $value = "";
    }
    else
    {
        $value = $this->diafan->formate_in_datetime($value);
    }
    break;

```

void **get_languages()** – записывает в глобальную переменную *\$this->diafan->languages* массив языков сайта.

- access: public

Пример:

```

/* modules/shop/shop.yandex.php
При формировании импорта для Яндекс.Маркета используется для определения языка.
*/
$this->diafan->get_languages();

```

Подключение шаблонов

- [Методы](#)

Для работы с шаблонами сайта в файле *includes/template.php* описан класс *Template*. В модуле объект этого класса доступен через переменную `$this->diafan->_tpl`, в шаблоне модуля объект класса доступен через переменную `$this`.

Методы

mixed `get(string $name, string $module, string $result)` – подключает шаблон / **`$module/views/$name.view.php`**

- access: public
- string `$name`: имя функции
- string `$module`: название модуля
- mixed `$result`: передаваемые в шаблон данные

Пример:

```
/* подключаем шаблон первой страницы модуля «Магазин» из контроллера в файле  
modules/shop/shop.php */  
$this->diafan->_tpl->get('first_page', 'shop', $result);  
  
/* подключаем шаблон формы авторизации в шаблоне модуля «Оформление заказа» в  
файле modules/cart/views/cart.view.form.php */  
echo $this->get('show_login', 'registration', $result);
```

void `htmleditor(string $text)` – заменяет шаблонные теги, ссылки в тексте, результат выводит в браузер.

- access: public
- string `$text`: исходный текст

Пример:

```
/* обрабатываем шаблонные теги в тексте из шаблона списка объявлений  
файл modules/ads/views/ads.view.list.php */  
$this->htmleditor($result['text']);
```

Пользователи

- [User](#)
- [Методы](#)

Для работы с пользователями определен класс *User* в файле *includes/user.php*.

Объект класса доступен в модулях как `$this->diafan->_user`.

User

Работа с пользователями в пользовательской части.

Свойства

boolean **\$admin**

- var: пользователь является администратором
- access: public

Пример:

```
/*
Воспользуемся проверкой свойства admin для вывода приветствия администратору
*/
if($this->diafan->_user->admin)
{
    echo 'Здравствуй, администратор!';
}
```

array **\$roles**

- var: права доступа администратора
- access: public

Используется в методе `roles($action, $module_name = "", $roles = array(), $type = 'admin')`

array **\$module_roles**

- var: права доступа для модулей
- access: public

Пример:

```
/* modules/useradmin/admin/useradmin/useradmin.admin.ajax.php
Определяет доступность модуля для данного пользователя
*/
```



```

if (! $this->diafan->_user->roles('edit', 'useradmin', $this->diafan->_user->module_roles) || !
$this->diafan->_user->roles('edit', $module_name))
{
    $this->result["errors"][0] = 'ERROR';
    return $this->send_errors(true);
}

```

boolean **\$checked**

- var: пользователь проверен по идентификационному хэшу
- access: public

Пример:

```

/* adm/includes/ajax.php
При отправке форм производится проверка на уникальность каждого запроса
посредством хэш-кода пользователя
*/
if (!$this->diafan->_user->checked)
{
    echo 'ERROR_HASH';
    return;
}

```

string **\$errauth**

- var: ошибка авторизации
- access: public

Пример:

```

/* modules/registration/registration.model.php
При ошибке авторизации содержит в себе константу данной ошибки
*/
switch ($this->diafan->_user->errauth)
{
    case 'wrong_login_or_pass':
        $result["error"] = $this->diafan->_('Неверный логин или пароль.', false);
        break;
    case 'blocked_30_min':
        $result["error"] = $this->diafan->_('Вы превысили количество попыток, поэтому будете
заблокированы на 30 минут', false);
        break;
    case 'blocked':
        $result["error"] = $this->diafan->_('Логин не активирован или заблокирован.', false);
        break;
    default:
        $result["error"] = "";
}

```

string **\$hash**

- var: идентификационный хэш
- access: private

Пример:

/ includes/user.php*

Свойство типа private. Используется только для внутренних нужд класса, в данном примере – для генерации уникального хэш-кода

**/*

```
if($this->hash)
{
    return $this->hash;
}
if ($this->id)
{
    $pass = DB::query_result("SELECT password FROM {users} WHERE id=%d LIMIT 1", $this->id);
    $this->hash = md5(substr($pass, mt_rand(0, 32), mt_rand(0, 32)).mt_rand(23, 567).substr($pass, mt_rand(0, 32), mt_rand(0, 32)));
    DB::query("INSERT INTO {sessions_hash} (user_id, created, hash) VALUES (%d, %d, '%h')", $this->id, time(), $this->hash);
    DB::query("DELETE FROM {sessions_hash} WHERE created<%d", time() - 7200);
    return $this->hash;
}
return "";
```

integer \$id = 0

- var: идентификатор пользователя
- access: public

Пример:

/ modules/usersettings/usersettings.php*

Подобная конструкция используется очень часто, когда нужно определить, авторизован ли пользователь на сайте или нет. В данном примере «гость» будет перенаправлен на страницу ошибки 403.

**/*

```
if(! $this->diafan->_user->id)
{
    include_once(ABSOLUTE_PATH.'includes/403.php');
}
```

array \$fields = array('name', 'fio', 'mail', 'created', 'role_id', 'lang_id', 'htmleditor', 'background')

- var: характеристики пользователя
- access: public

Пример:

```
/* modules/registration/registration.model.php
```

Позволяет получить доступ к данным авторизованного пользователя. В примере присваивается ФИО пользователя, которое будет выведено в форме авторизации

```
*/
```

```
$result["fio"] = $this->diafan->_user->fio;
```

string **\$session**

- var: данные, хранящиеся в сессии
- access: public

Пример:

```
/* includes/session.php
```

Хранит данные о сессии пользователя, для неавторизованного посетителя равна "

```
*/
```

```
if ($user && $user->id > 0)
{
    $diafan->_user->set($user);
    return $user->session;
}
```

Методы

boolean **anonymous** ([string **\$session** = ""]) – определяет, что текущий пользователь не авторизован

- return: true
- access: public
- string **\$session**: данные, записанные в сессии

Пример:

```
/* includes/session.php
```

Если пользователь не авторизован, то в таблицу sessions заносится запись с user_id = 0

```
*/
```

```
$session = DB::query_result("SELECT session FROM {sessions} WHERE session_id = '%s'
AND hostname='%s' AND user_agent='%s' LIMIT 1", $key, $_SERVER['REMOTE_ADDR'],
$_SERVER['HTTP_USER_AGENT']);
```

```
$diafan->_user->anonymous($session);
return $session;
```

boolean **auth** (**\$form_values**) – проверяет, авторизован ли пользователь.

- access: public
- **\$form_values**

Пример:

```

/* includes/init.php
Проверка данных при авторизации пользователя
*/
if (! empty($_POST['action']) && $_POST['action'] == 'auth')
{
    $this->_user->auth($_POST);
}

```

boolean **auth (\$form_values)** – проверяет, авторизован ли пользователь.

- access: public
- \$form_values

Пример:

```

/* includes/init.php
Проверка данных при авторизации пользователя с возможностью авторизации через
сервис loginza
*/
if (! empty($_POST['action']) && $_POST['action'] == 'auth')
{
    $this->_user->auth($_POST);
}
elseif (! empty($_POST['token']))
{
    $this->_user->auth_loginza();
}

```

string **get_hash()** – генерируем идентификационный пользовательский хэш.

- access: public

Пример:

```

/* includes/ajax.php
Используется для проверки уникальности запроса для данного пользователя
*/
$this->result["hash"] = $this->diafan->_user->get_hash();

```

void **delete_session_hash(\$id)** – удаляем уникальный хэш сессии.

- access: public
- integer \$id: номер хэша

Пример:

```

/* includes/user.php
Служебная функция. Используется непосредственно в самом классе.
*/
if ($id = DB::query_result("SELECT id FROM {sessions_hash} WHERE user_id=%d AND

```

```

created>%d AND hash='%h' LIMIT 1", $this->id, time() - 7200, $_POST["check_hash_user"])))
{
    register_shutdown_function(array($this, 'delete_session_hash'), $id);
    $this->checked = true;
}

```

void **create_avatar(\$name, \$file)** – загружает аватар.

- access: private
- string \$name: логин пользователь
- string \$file: файл аватара

Пример:

```

/* includes/user.php
Служебная функция. Используется непосредственно в самом классе.
*/
if (!empty($profile->photo))
{
    $this->create_avatar($name, $profile->photo);
}

```

boolean **_log()** – проверяет лог авторизации для блокировки попытки подбора паролей.

- access: private

Пример:

```

/* includes/user.php
Служебная функция. Если превышено число попыток входа, то ip блокируется на 30 минут.
*/
if ($this->_log())
{
    $this->errauth = 'blocked_30_min';
    return false;
}

```

boolean **update_log()** – обновляет лог авторизации для блокировки попытки подбора паролей.

- access: private

Пример:

```

/* includes/user.php
Служебная функция. Собирает информацию о неудачных попытках входа в систему.
*/
$this->update_log();
$this->errauth = 'wrong_login_or_pass';

```

```
return false;
```

boolean **logout()** – очищает информацию о текущем пользователе.

- return: true
- access: public

Пример:

```
/* includes/init.php  
Инициализирует выход пользователя из системы  
*/  
if (strpos($_GET["rewrite"], "logout") !== false)  
{  
    $this->_user->logout();  
}
```

boolean **roles (string \$action, [string \$module_name = ''], [array \$roles = array()], [string \$type = 'admin'])** – проверяет есть ли права у пользователя на действие для модуля.

- return: true
- access: public
- string \$action: действие
- string \$module_name: модуль
- array \$roles: права пользователя
- string \$type: часть сайта административная/пользовательская

Пример:

```
/* adm/includes/del.php  
Проводит проверку прав пользователя на операцию удаления для данного модуля  
*/  
if (! $this->diafan->_user->roles('del', $this->diafan->rewrite))  
{  
    echo "{error:'ROLES'}";  
    return;  
}
```

boolean **set (object \$user)** – определяет текущего пользователя.

- return: true
- access: public
- object \$user: данные о текущем пользователе

Пример:

```
/* includes/session.php  
Если авторизация пользователя прошла успешно, то назначить его текущим  
пользователем  
*/
```

```
if ($user && $user->id > 0)
{
    $diafan->_user->set($user);
    return $user->session;
}
```

Почта

- [Пример использования](#)

Работа с электронной почтой основана на плагине [PHPMailer](#). Соединение diafan.CMS и плагина обеспечивает функция `send_mail`, описанная в файле `includes/mail.php`. В CMS данная функция используется в основном для уведомлений пользователей о каких-либо действиях, например, о покупке товара (`modules/cart/cart.ajax.php`).

boolean **send_mail**(string|array \$recipient, string \$subject, string \$body, [string \$from = ""]) – отправляет электронное письмо.

- string|array \$recipient: получатель/получатели
- string \$subject: тема письма
- string \$body: содержание письма
- string \$from: адрес отправителя

Пример:

```
// Отправляем Васе уведомление на почтовый ящик, каждый раз при просмотре его
// товара
// файл modules/shop/shop.model.php метод id()
$sent = send_mail(
    "vasya@mail.ru", // кому
    "Кто-то смотрит твой товар на сайте ".BASE_PATH_HREF, // тема
    "Привет, кто-то зашел на страницу твоего товара
    ".BASE_PATH_HREF.$this->diafan->_route->current_link(), // текст сообщения
); // === true, если письмо успешно отправлено
```

Пример использования

Для примера в магазине реализуем кнопку «Порекомендовать другу», при нажатии на которую появляется форма с полем ввода email.

Пример:

Создадим файл `modules/shop/views/shop.view.email_form.php`

```
<div class="modal-wrap" id="shop-modal-wrap">
  <div class="modal" id="shop-modal">
    <div class="window" style="width:300px">
      <div class="window_title">
        Послать другу
      </div>
      <form method="POST" enctype="multipart/form-data" action="" class="ajax">
        <input type="hidden" name="module" value="shop">
        <input type="hidden" name="action" value="mail">
        <input type="hidden" name="good_id" value="".$result['id']. ">
        <input type="hidden" name="ajax" value="0">
```



```

        <table style="width:300px">
        <tr>
        <td>E-mail:</td>
        <td>
            <input type="email" name="mail">
            <div class="errors error_mail" style="display:none"></div>
        </td>
        </tr>
        </table>
        <input type="submit" value="Отправить">
    </form>
    <div class="errors error"></div>
    <div class="modal-close close"></div>
</div>
</div>
</div>

```

В файле *modules/shop/views/shop.view.id.php* в напомним:

```
$this->get('email_form', 'shop', $result);
```

Теперь AJAX обработчик *modules/shop/shop.ajax.php*

```

switch ($_POST['action'])
{
    case 'mail':
        return $this->send_friends();
        // ...
        private function send_friends()
        {
            if (empty($_POST['mail']))
            {
                $this->result['errors']['mail'] = 'Пожалуйста, введите e-mail';
            }
            else
            {
                $this->valid_email($_POST['mail'], "mail");
            }
            $row = DB::fetch_array(DB::query("SELECT [name],site_id,cat_id,id FROM {shop}
            WHERE id=%d LIMIT 1", $_POST['good_id']));
            if (empty($row))
            {
                $this->result['errors'][0] = 'Ошибка отправки запроса';
            }
            $row['link'] = BASE_PATH_HREF.$this->diafan->_route->link($row["site_id"], "shop",
            $row["cat_id"], $row["id"]);
            if ($this->send_errors())
                return true;
            require_once (ABSOLUTE_PATH.'includes/mail.php');
            $this->result['errors'][0] = send_mail($_POST['mail'], "Рекомендую посмотреть",
            'Здравствуйте! Ваш друг рекомендует Вам посмотреть <a href="'. $row['link']. ">'.
            $row['name'].", $this->diafan->configmodules("emailconf", 'shop') ? $this->diafan-

```

```
>configmodules("email", 'shop') : ") ? 'Отправлено' : 'Ошибка';  
    return $this->send_errors();  
    }  
}
```

Режим разработки

- [Методы](#)

В diafan.CMS предусмотрены специальные инструменты для разработчика: режим разработки (отображает все PHP-ошибки) и профилирование запросов.

Они описаны в классе *Dev*, в файле *includes/developer.php*.

Методы

boolean **get_profiling()** – профилирование запросов.

- access: public

boolean **set_error()** – разрешает/запрещает вывод ошибок.

- access: public

boolean **set_profiling()** – активирует профилирование запросов, если это разрешено в параметрах.

- access: public

Сессии

- [Методы](#)

Для работы с сессией описан класс *Session* в файле *includes/session.php*.

Внимание! Методы данного класса являются системными. Их изменение может вызвать нестабильную работу всей системы.

Методы

boolean **close()** – закрывает сессию и освобождает ресурсы.

- return: true
- access: public

boolean **destroy(string \$key)** – удаляет сессию.

- return: true
- access: public
- string \$key: идентификатор сессии

boolean **gc([integer \$lifetime = 1814400])** – чистит мусор – удаляет сессии старше \$lifetime.

- return: true
- access: public
- integer \$lifetime: время хранения сессии в секундах

boolean **init()** – стартует сессию.

- return: true
- access: public

boolean **open(string \$save_path, string \$session_name)** – открывает сессию.

- return: true
- access: public
- string \$save_path
- string \$session_name

string **read(string \$key)** – читает сессию.

- return: сериализованные данные сессии
- access: public
- string \$key: идентификатор сессии

boolean **write(string \$key, string \$value)** – записывает данные в сессию.

- return: true
- access: public
- string \$key: идентификатор сессии
- string \$value: сериализованные данные сессии

boolean **duration()** – определяет продолжительность сессии.

- return: true
- access: public

Сжатие страниц

- [Методы](#)

Для обеспечения сжатия страниц (если это поддерживается обозревателем) в файле *includes/gzip.php* определен системный класс *Gzip*, который автоматически инициализируется при каждом обращении к странице.

Внимание! Данный класс является системным и не может быть использован где-то еще.

Методы

boolean **do_gzip()** – выдает сжатые данные, очищает (стирает) буфер вывода и отключает буферизацию вывода.

- return: true

boolean **init_gzip()** – иницирует сжатие. Включается буферизация вывода, определяются и передаются необходимые HTTP заголовки.

- return: true

Установка модулей

- [Методы](#)

Для установки модулей в файле *includes/install.php* определен класс *Install*.

При создании экземпляра класса нужно передать массив с идентификаторами языков сайта.

Пример:

```
$langs = array(1, 2);  
$install = new Install($langs);
```

[Описание файла установки модуля.](#)

Внимание! Данный класс является системным и не может быть использован где-то еще.

Методы

void install (string \$module, boolean \$install_example) – устанавливаем модуль.

- access: public
- string \$module: название модуля
- boolean \$install_example: установить примеры

Пример:

```
/* modules/update/admin/update.admin.install.php  
Добавление модуля в административную панель  
*/  
$inst->install($module, (!empty($_POST["example_yes"])));  
//установка прав на административную часть установленного модуля текущему  
пользователю  
if(! $this->diafan->_user->roles('all', 'all'))  
{  
    $result = DB::query("SELECT rewrite FROM {adminsitedata} WHERE rewrite='%s' OR rewrite  
LIKE '%s%%'", $module, $module);  
    while ($row = DB::fetch_array($result))  
    {  
        DB::query("INSERT INTO {users_role_perm} (role_id, perm, rewrite, type) VALUES(%d,  
'all', '%s', 'admin')", $this->diafan->_user->role_id, $row["rewrite"]);  
    }  
}
```

void uninstall (string \$module) – удаляет модуль.

- access: public
- string \$module: название модуля

Пример:

```
/* modules/update/admin/update.admin.install.php  
Удаление пользователем модуля из административной панели  
*/  
if (empty($_POST[$module]) && $ins)  
{  
    $inst->uninstall($module);  
}
```


Файлы

- [Методы](#)

Для работы с файлами и папками в файле *includes/files.php* определен класс *Files*.

Методы

boolean **check_file (string \$file_path)** – проверяет существует ли файл.

- access: public
- string \$file_path: путь до файла относительно корня сайта

Пример:

```
if(Files::check_file("cache/file.tmp")) { echo "Файл существует"; }
```

void **upload_file (string \$tmp_path, string \$name, string \$path)** – загружает файл.

- access: public
- string \$tmp_path: путь, где храниться временный файл
- string \$name: имя файла
- string \$path: путь до папки, в которую будет загружен файл, относительно корня сайта

Пример:

```
Files::upload_file($_FILES["image"]["tmp_name"], "new_image.jpg", "userfiles/images");
```

void **save_file (string \$content, string \$file_path)** – сохраняет файл.

- access: public
- string \$content: содержание файла
- string \$file_path: путь до файла относительно корня сайта

Пример:

```
Files::save_file($_POST["content"], "cache/file.tmp");
```

void **rename_file (string \$name, string \$old_name, string \$path)** – переименовывает файл.

- access: public
- string \$name: новое имя
- string \$old_name: старое имя
- string \$path: путь до папки, в которой лежит файл, относительно корня сайта

Пример:

```
Files::rename_file("file_new.tmp", "file.tmp", "cache");
```

void **delete_file** (string \$file_path) – удаляет файл.

- access: public
- string \$file_path: путь до файла относительно корня сайта

Пример:

```
Files::delete_file("file.tmp");
```

boolean **check_dir** (string \$dir_path) – проверяет существует ли папка.

- access: public
- string \$dir_path: путь до папки относительно корня сайта

Пример:

```
Files::check_dir("cache/files");
```

void **create_dir** (string \$name, string \$path) – создает папку.

- access: public
- string \$name: имя новой папки
- string \$path: путь до папки-родителя относительно корня сайта

Пример:

```
Files::create_dir("files", "cache");
```

void **rename_dir** (string \$name, string \$old_name, string \$path) – переименовывает папку.

- access: public
- string \$name: новое имя папки
- string \$old_name: старое имя папки
- string \$path: путь до папки-родителя относительно корня сайта

Пример:

```
Files::rename_dir("images", "files", "cache");
```

void **delete_dir** (string \$dir_path) – удаляет папку.

- access: public
- string \$dir_path: путь до папки относительно корня сайта

Пример:

```
Files::delete_dir("cache/files");
```

boolean **is_writable** (string \$path, [boolean \$ftp = false]) – определяет, доступны ли файл или папка для записи.

- access: public
- string \$path: путь до файла или папки относительно корня сайта
- boolean \$ftp: учитывать возможность редактирования по FTP

Пример:

```
if(Files::is_writable("cache/files")) { echo "папка доступна для записи"; }
```

SMS

- [Методы](#)

Для отправки SMS в файле *includes/sms.php* описан класс *Sms*.

Рассылка SMS подключается и настраивается в [Параметрах сайта](#) и используется в разных модулях.

Методы

`void send(string $text, string $to)` – отправляет SMS.

- access: public
- string \$text: текст SMS
- string \$to: номер получателя

Ниже приведен пример использования данного метода.

Пример:

```
/* modules/faq/faq.ajax.php
Отправка СМС-сообщения администратору сайта о том, что задан новый вопрос.
*/
private function send_sms()
{
    if (! $this->diafan->configmodules("sendsmsadmin", 'faq', $this->site_id))
        //если модуль не подключен
        return false;
    $message = $this->diafan->configmodules("sms_message_admin", 'faq', $this->site_id);
    //текст сообщения, взятый из настроек
    $to = $this->diafan->configmodules("sms_admin", 'faq', $this->site_id);
    //номер телефона администратора
    include_once ABSOLUTE_PATH.'includes/sms.php';
    Sms::send($message, $to); // отправка сообщения
}
```

`string|boolean false validate_to(string $to)` – валидация номера получателя.

- access: public
- string \$to: номер получателя

Метод используется внутри класса, его основная задача – проверка номера абонента перед отправкой сообщения.

Пример:

```
/* includes/sms.php
```

Проверка номера на отсутствие иных знаков кроме цифр.

*/

```
$to = preg_replace('/[^0-9]+/', '', $to);
```

```
if($error = self::validate_to($to))
```

```
{
```

```
    return $error;
```

```
}
```

Список внешних плагинов

Плагины – функционал сторонних разработчиков, используемый для выполнения определенных задач.

diafan.CMS использует следующие плагины:

- [PHPMailer](#) – работа с электронной почтой;
- [Kohana](#) – заимствована работа с кодировкой UTF8;
- [FTP от David Grudl](#) – доступ к FTP серверу;
- [JsHttpRequest](#) – заимствовано преобразование данных в формат JSON;
- [KCAPTCHA](#) – работа с защитным кодом;
- [TinyMCE](#) – визуальный редактор;
- [Image Manager](#) – плагин для TinyMCE, обеспечивающий загрузку файлов;
- [ArtLebedevStudio.RemoteTypograf](#) – типограф от студии Артемия Лебедева;
- [IDNA Convert](#) – Punycode-конвертация доменов;
- [Lingua_Stem_Ru](#) – Стеммер Портера;
- [jQuery](#) – JavaScript библиотека;
- [jQuery Form Plugin](#) – jQuery плагин для работы с формами;
- [prettyPhoto](#) – jQuery плагин для анимации открытия изображений;
- [jQuery Tooltip plugin](#)
- [Multiple file upload component with progress-bar, drag-and-drop](#);
- [jQuery UI](#);
- [Superfish](#);
- [hoverIntent by Brian Cherne](#);
- [jCarouselLite](#);
- [imgAreaSelect](#);
- [timePicker](#).

Практические примеры

- [Как создать свой модуль](#)
- [Как добавить шаблонный тег](#)
- [Как добавить новое поле в модуль](#)
- [Как получить свободные данные из БД и вывести их](#)
- [Как вставить сторонний PHP-код](#)
- [Как сделать зависимые Ajax-списки](#)
- [Как в модуле использовать другой модуль](#)
- [Переопределение функционала в административной части модуля](#)
- [Кастомизация функций](#)
- [Изменение административной части](#)
- [Как в поиске по товарам заменить чекбоксы на список](#)
- [Как вывести дополнительную характеристику товара в любом месте](#)
- [Как вывести список товаров в корзине в шаблонном теге](#)
- [Как добавить виджет комментариев из социальной сети](#)

Как создать свой модуль

- [Создаем таблицу в БД](#)
- [Панель администрирования](#)
- [Вкладки для части модуля и конфигурация](#)
- [Пользовательская часть](#)
- [Шаблонный тег](#)

В данном разделе документации рассказано как создать модуль для diafan.CMS. На первый взгляд это легко, но, как и везде, есть свои особенности. Приложение к данному разделу – каркас модуля, который можно [скачать по этой ссылке](#).

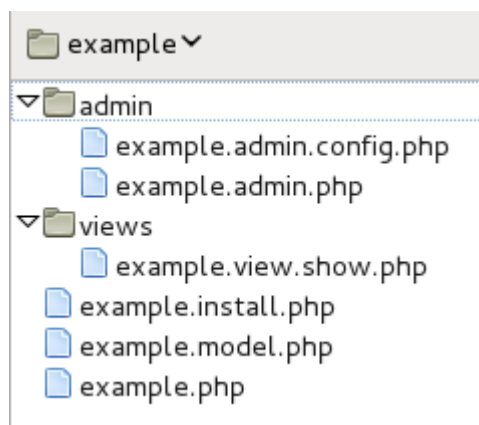


Рис. 1. Структура модуля diafan.CMS

В папке *admin* находится административная часть модуля: редактирование данных модуля *example.admin.php* и настройки модуля *example.admin.config.php*. В корне расположены файлы *example.model.php* – *example.view.php* – папка *views* «Модель-представление-контроллер». Файл *example.install.php* – установка модуля.

[Цитата из wikipedia](#)

Концепция *MVC* позволяет разделить данные, представление и обработку действий пользователя на три отдельных компонента:

- *Модель* (англ. *Model*). Модель предоставляет знания: данные и методы работы с этими данными, реагирует на запросы, изменяя своё состояние. Не содержит информации, как эти знания можно визуализировать.
- *Представление, вид* (англ. *View*). Отвечает за отображение информации (визуализация). Часто в качестве представления выступает форма (окно) с графическими элементами.
- *Контроллер* (англ. *Controller*). Обеспечивает связь между пользователем и системой: контролирует ввод данных пользователем и использует модель и представление для реализации необходимой реакции.

Для того чтобы подключить наш сырой модуль к diafan.CMS, необходимо сделать записи в таблицах [{modules}](#) и [{adminsite}](#). Для этого в файле *example.install.php* прописываем:

Пример:

```
// Название модуля для списка модулей при установке
$title = "Название модуля";

// SQL-запросы к базе данных, выполняемые при установке
$db = array(
    // делаем запись в таблице {modules}
    "modules" => array(
        array(
            // название подмодуля
            "name" => "example",
            // название модуля
            "module_name" => "example",
            // имеет пользовательскую часть
            "site" => 1,
            // можно прикрепить к странице сайта
            "site_page" => 1,
            // имеет административную часть
            "admin" => 1,
            // название
            "title" => $title,
        )
    ),
    // делаем запись в таблице {adminsites}
    "adminsites" => array(
        array(
            // название пункта меню в административной части
            'name' => $title,
            // ЧПУ (должно совпадать с названием файла)
            'rewrite' => 'example',
            // часть меню
            'group_id' => 1,
            // порядок следования
            'sort' => 8,
            // пункт отображается в меню
            'act' => 1,
            // вложенные страницы
            'children' => array (
                array (
                    'name' => 'Настройки',
                    'rewrite' => 'example/config',
                    'sort' => 1,
                    'act' => 1,
                )
            ),
        ),
    ),
);
```

Теперь модуль появится в списке модулей для установки в разделе «Модули и БД» – «Установка модулей». Устанавливаем модуль.

Следующим шагом будет добавление нашего модуля на какую-нибудь страницу сайта. Идем в админку, создаем обычную текстовую страницу и во вкладке «Дополнительно» прикрепляем к ней наш новый модуль.

Все, модуль готов к использованию!

Правда пока он ничего не делает, поэтому давайте сделаем его функциональным.

Давайте сделаем модуль, который будет позволять размещать объявления на сайте.

Внимание!

Все что вы прочитаете ниже не жесткие пошаговые инструкции и выполнять действия в том порядке, котором это сделано здесь не обязательно. У каждого программиста свой подход к решению поставленной задачи и необходимо уметь самостоятельно думать, чтобы все получалось.

Создаем таблицу в БД

Для начала определимся с задачами, которые должен выполнять модуль:

- позволять зарегистрированным пользователям сайта добавлять объявления
- позволять администратору редактировать объявления
- позволять администратору размещать объявления

Объявления будут размещены в базе данных в таблице *prefix_example*, и нам необходимо разработать ее структуру.

Внимание!

Префикс по умолчанию – *diafan_*. Вы могли переименовать его при установке diafan.CMS, поэтому создавайте таблицу по подобию других таблиц в БД.

Для работы с БД можно использовать [phpMyAdmin](#), который обычно есть на хостинге.

	Поле	Тип	Сравнение	Атрибуты	Null	По умолчанию	Дополнительно
☐	id	int(11)		UNSIGNED	Нет	Нет	AUTO_INCREMENT
☐	user_id	int(11)		UNSIGNED	Нет	0	
☐	text	text	utf8_general_ci		Нет	Нет	
☐	created	int(12)			Нет	0	

- **user_id** – id нашего зарегистрированного пользователя, который добавил объявление
- **created** – время создания
- **text** – текст объявления

Таблица с информацией о пользователях у нас уже есть, это модуль по умолчанию [«Пользователи»](#), поэтому мы будем использовать эту информацию. Нам достаточно хранить только идентификатор пользователя в поле *user_id* нашей новой таблицы.

Таблицу описываем в файле *example.install.php*, чтобы можно было использовать модуль повторно.

Пример:

```
$db = array(
    ...
    // основная таблица модуля, в которой хранятся объявления
    "tables" => array(
        array(
            "name" => "example",
            "fields" => array(
                array(
                    "name" => "id",
                    "type" => "INT(11) UNSIGNED NOT NULL AUTO_INCREMENT",
                ),
                array(
                    "name" => "user_id",
                    "type" => "INT(11) UNSIGNED NOT NULL DEFAULT '0'",
                ),
                array(
                    "name" => "created",
                    "type" => "INT(10) UNSIGNED NOT NULL DEFAULT '0'",
                ),
                array(
                    "name" => "text",
                    "type" => "text NOT NULL DEFAULT ''",
                ),
            ),
            "keys" => array(
                "PRIMARY KEY (id)",
            ),
        ),
    ),
);
```

Панель администрирования

Начнем разработку нашего модуля с админки, перед этим не забыв включить [«Режим разработки»](#) в параметрах сайта, чтобы видеть лог наших возможных ошибок.

Внимание!

Обязательно включайте режим разработки и отключайте кэширование в настройках сайта при любых правках любых модулей.

Если открыть админку нашего нового модуля, мы ничего не увидим, потому что мы ничего не описали. Когда административная часть diafan.CMS подключает [администрирование модуля](#), то ядро системы сразу автоматически подключает для управления информацией модуля все [действия](#). В модуле для начала достаточно просто описать, что и как надо выводить и что редактировать.

Откроем файл *example.admin.php* и опишем класс *Example_admin*:

Пример:

```
class Example_admin extends Frame_admin
{
    // название таблицы бд, по которой будет формироваться список
    public $table = 'example';
    // описание полей таблицы
    public $variables = array (
        'main' => array (
            'created' => array(
                'type' => 'datetime',
                'name' => 'Дата создания',
            ),
            'user_id' => array(
                'type' => 'select',
                'name' => 'Автор',
            ),
            'text' => array(
                'type' => 'editor',
                'name' => 'Текст объявления',
            ),
        ),
    );
    // функция, которая определяет что выводит модуль при открытии
    public function show()
    {
        // ссылка на добавление нового объявления
        $this->diafan->addnew_init('Добавить объявление');
        // список объявлений
        $this->diafan->list_row();
    }
}
```

Свойство класса [\\$table](#) определяет в какой таблице будут храниться наши объявления, массив [\\$variables](#) описывает каждое поле таблицы, функция *show()* выводит список объявлений.

Описывая поля, мы показываем diafan.CMS, что мы хотим править и в каком виде. Заметьте, что порядок следования пунктов повторяется и на странице в админке. Для редактирования пользователя удобнее сделать выпадающий список, где были бы имена зарегистрированных пользователей. Для этого задаем дополнительное свойство [\\$select](#), в котором указываем, в какой таблице базы данных и какие данные брать.

Пример:

```
public $select = array(
    'user_id' => array(
        'users',
        'id',
    ),
)
```

```
'fio',
'',
'',
"trash='0'",
),
);
```

И последнее. Мы должны указать, что основное поле у нас называется text (по умолчанию name):

Пример:

```
public $text_for_base_link = array('variable' => 'text');
```

Все, этого вполне достаточно чтобы выводить список объявлений, добавлять, редактировать и сохранять их.

Теперь зададим некоторые настройки с помощью свойства `$config`. Добавим возможность удаления объявления и вывод даты и времени добавления в списке.

Пример:

```
public $config = array (
    'del', // удалить
    'datetime', // показывать дату в списке, сортировать по дате
);
```

Для пущей красоты и информативности давайте добавим дополнительное поле «пользователь», которому объявление принадлежит.

Пример:

```
public $config_other_row = array('user_id' => 'string');
```

Ура, теперь у нас в списке выводится объявление, время добавления и id пользователя.

 [Добавить объявление](#)

☐	23.10.2012 14:34	Куплю квартиру	1
--------------------------	------------------	--------------------------------	---

☐	23.10.2012 14:34	Продам дом	1	
--------------------------	------------------	----------------------------	---	---

☐ Отметить все

Как вы видите, функция `list_row` выводит поля из базы данных в том виде, в котором они в нее записаны, но это можно легко исправить!

Напишем простую функцию, переопределяющую системное отображение. Формат названия

функции таков: *other_row_переменная(\$row)*, где *\$row* – массив значений для текущего элемента:

Пример:

```
public function other_row_user_id($row)
{
    return '<td>'.DB::query_result("SELECT name FROM {users} WHERE id=%d",
    $row['user_id']).'</td>';
}
```

То есть, если diafan.CMS обнаруживает эти функции с названием по маске *other_row_**, она понимает, что поле *user_id* нужно выводить так, как описано в этой функции.

Результат налицо

Добавить объявление

☐	23.10.2012 14:34	Куплю квартиру	Иван Иванович	
☐	23.10.2012 14:34	Продам дом	Иван Иванович	

☐ Отметить все Удалить Применить

Идем далее. Расширим функциональность нашего модуля, чтобы наши объявления сортировались, удалялись и активировались по нашему велению.

В базу данных добавим поля удаления в корзину, активности и сортировки:

Поле	Тип	Сравнение	Атрибуты	Null	По умолчанию	Дополнительно
id	int(11)		UNSIGNED	Нет	Нет	AUTO_INCREMENT
user_id	int(11)		UNSIGNED	Нет	0	
text	text	utf8_general_ci		Нет	Нет	
created	int(12)			Нет	0	
trash	enum('0','1')	utf8_general_ci		Нет	0	
act	enum('0','1')	utf8_general_ci		Нет	0	
sort	int(11)		UNSIGNED	Нет	0	

И пропишем настройки в *\$config*.

Пример:

```
public $config = array('del', 'act', 'order', 'trash');
```

Теперь у нас в списке объявлений появятся иконки удаления, активности, и их можно будет сортировать.

Внимание!

Также не стоит забывать про переменную [\\$where](#), которая позволяет нам добавлять собственные параметры к выборке функции *list_row*.

Давайте с ее помощью для примера организуем перед списком фильтр объявлений по имени пользователя.

Сделаем простую форму с GET запросом.

Пример:

```
public function show()
{
    $this->diafan->addnew_init('Добавить объявление');
    echo '<form method="GET">'. $this->diafan->_('Пользователь').':
    <select name="user"><option value="">'. $this->diafan->_('Все'). '</option>';
    $result = DB::query("SELECT id, fio FROM {users} WHERE trash='0' ORDER BY fio ASC");
    while($user = DB::fetch_array($result))
    {
        echo '<option value="'. $user["id"].">'. $user["fio"]. '</option>';
    }
    echo '</select><input type="submit" value=" OK "></form>';
    $this->diafan->list_row();
}
```

Свойство будем определять на лету, поэтому воспользуемся функцией [prepare_config\(\)](#).

Пример:

```
public function prepare_config()
{
    if(! empty($_GET['user']))
    {
        $this->diafan->where = " AND user_id='". $this->diafan->get_param($_GET, 'user', "", 2)."'";
    }
}
```

В общем, вид представления административной части модуля зависит лишь от вашей фантазии. Никто не принуждает вас использовать функцию *list_row*, вы можете написать любой собственный обработчик.

Внимание!

Мощным средством расширения функционала модуля является тип **function**.

Он позволяет определять пользовательские функции для редактирования и сохранения поля. Давайте сделаем так, чтобы при редактировании *user_id* имя пользователя у нас было не просто выпадающим списком, а выбиралось с помощью AJAX-запроса:

Переназначим *\$variables*, укажем для *user_id* тип не *select*, а *function*.

Пример:

```
public $variables = array (  
    'main' => array (  
        ...  
        'user_id' => array(  
            'type' => 'function',  
            'name' => 'Автор',  
        ),  
        ...  
    ),  
);
```

И объявим две функции

Пример:

```
public function edit_variable_user_id(){}  
  
public function save_variable_user_id(){}
```

Займемся визуальным отображением редактирования поля *user_id* и будем править функцию *edit_variable_user_id*. Выведем то, что нам передается в функцию, и посмотрим, что есть там интересного.

Пример:

```
print_r($this->diafan);  
  
// параметр id редактируемого элемента передаваемый в URL  
[edit] => 2  
  
// все значения этого поля из таблицы  
[values] => Array  
(  
    [id] => 2  
    [user_id] => 2  
    [text] => Продам квартиру  
    [created] => 1322127780  
    [trash] => 0  
    [act] => 1  
    [sort] => 2  
)  
  
// текущее поле и его значение из базы данных  
[key] => user_id  
[value] => 2
```

Раздолья для фантазии много, но мы напишем простой jquery обработчик, отправляющий AJAX-запрос

Пример:

```
public function edit_variable_user_id()
{
    echo '
    <script type="text/javascript">
    $(document).ready(function(){
        $(".user_id").click(function(){
            $.ajax({
                url : window.location.href,
                type : "POST",
                dataType : "json",
                data : {
                    module: "example",
                    action: 1,
                    user_id: $(this).attr("rel")
                },
                success : (function(response)
                {
                    alert(response.name);
                })
            });
        });
    });
    </script>
    <tr>
        <td class="td_first">Нажми</td>
        <td><div class="user_id" rel="".$this->diafan->value."><b>ЗДЕСЬ</b></div></td>
    </tr>';
}
```

Обработчик по нажатию на элемент с классом *user_id* отправляет AJAX-запрос текущей странице с параметрами *module*, *action* и *user_id*.

Теперь нам нужно написать *example.admin.ajax.php* (скачайте [example.admin.function.zip](#)), в котором, собственно, и будет происходить обработка запроса. Файл *example.admin.ajax.php* подключается лишь тогда, когда одновременно отправляются POST переменные *action* и *module*. Значение переменной *module* должно соответствовать названию нашего модуля.

Часть кода AJAX обработчика (полный смотрите в примере)

Пример:

```
if (!empty($_POST['user_id']))
{
    $this->result["name"] = DB::query_result("SELECT name FROM {users} WHERE id=%d LIMIT 1", $_POST['user_id']);
}
...
$this->send_json();
```

Соответственно, в функции *save_variable_user_id* обрабатывается сохранение поля. Тут вы

можете вставлять данные в другую таблицу, обрабатывать самостоятельно переменные `$_POST`, дважды досчитать до бесконечности.

Для того чтобы записать данные в таблицу `{example}`, которую использует наш модуль, можно сделать такую хитрость:

Пример:

```
$this->diafan->set_query("user_id=%d");  
$this->diafan->set_value(1);
```

Таким образом, мы добавили к SQL запросу UPDATE поле `user_id` со значением 1.

Еще одной интересной переменной является массив `$this->diafan->oldrow`, в котором находится текущая выборка из базы данных.

Как вы могли сами увидеть, *function* предоставляет огромные возможности для расширения функционала административной части модуля, все зависит от вашей фантазии и правильного крепления рук к телу.

Еще есть замечательный тип в *\$variables* – *module*. Он означает, что необходимо подключить к редактированию сторонний модуль.

Как это работает.

Там где нужно подключить модуль *example*, в его *file.admin.php* в *\$variables* прописываем

Пример:

```
public $variables = array(  
    'main' => array(  
        'example' => 'module',  
        ...  
    ),  
    ...  
);
```

При таком раскладе, при редактировании элемента модуля *file* diafan.CMS будет искать файл

modules/example/admin/example.admin.inc.php, где описаны функции *edit()* и *save()*.

Это работает практически так же, как и тип *function*, только между модулями.

Вкладки для части модуля и конфигурация

В diafan.CMS административную часть модуля можно еще дополнять различными расширениями, вкладками (подразделами модуля, например, как заказы в магазине). Загляните в папку *modules/shop/admin*

Делается это так: создается файл *example.admin.bolt.php* и в нем инициализируется класс

```
class Example_admin_bolt extends Frame_admin
```

Потом мы идем в <http://site.ru/admin/adminsite/> и добавляем подраздел нашему модулю с псевдоссылкой *example/bolt*. Количество расширений модуля зависят лишь от вашей фантазии.

Есть еще особый тип расширения – **конфигурация**

Пример [example.admin.config.php](#) скачать можно [здесь](#).

Регистрируется он так же, как и обычное расширение: в <http://site.ru/admin/adminsite/> добавляем подраздел «Настройки» с псевдоссылкой *example/config*.

Основной его отличительной особенностью является настройка в `$config`.

```
public $config = array('config');
```

Если она установлена, то сохранение происходит не в таблицу, указанную в `$table`, а в таблицу `{config}`, для доступа к которой предусмотрена специальная функция ядра [configmodules](#), которую можно вызвать откуда угодно.

Пользовательская часть

Если мы подключили модуль к странице сайта, то при ее открытии из конструктора *example.php* вызывается функция `show_module()`. Как вы можете видеть из примера, при инициализации модуля подключается модель, результат исполнения записывается в переменную `$this->result`, а после в `show_module()` подключается файл *example.view.show.php*.

Функция контроллера (*example.php*) `init()` выполняется всегда при подключении модуля. В ней описывается логика выбора модели (*example.model.php*). Для начала рассмотрим самые базовые возможности.

Массив `rewrite_variable_names` – в нем описываются имена, которые будут считаны из URL как переменные.

Пример:

```
public function init()
{
    $this->rewrite_variable_names = array('page', 'show', 'cat');
    $this->diafan->rewrite_variable_names = $this->rewrite_variable_names;
    ...
    $this->get_global_variables();
}
```

Полный список доступных переменных смотрите [здесь](#)

Если очень нужна своя переменная, передаваемая через URL, или просто GET – это не кошерно, можно залезть в *includes/init.php* и дополнить массив `rewrite_variable_names_all`.

Шаблонный тег

Теперь модуль мы умеем выводить (функция *show_module*), а ведь есть еще такая тема, как [шаблонный тег](#), который можно вставить в основной шаблон сайта, либо прописать в контенте страницы сайта.

В документации уже есть раздел про то, [как создать свой шаблонный тег](#).

diafan.CMS не делает никаких ограничений на количество аргументов и название функции!

Давайте напишем функцию, которая будет нам выводить последние объявления.

Сначала идем в *example.php* – контроллер.

Пример:

```
public function show_last($attributes)
{
    // принимаем атрибут count
    $attributes = $this->get_attributes($attributes, 'count');
    // подключаем модель
    include_once(ABSOLUTE_PATH.'modules/example/example.model.php');
    // получаем результат
    $model = new Example_model($this->diafan);
    // не забываем о правиле ВСЕ ПЕРЕМЕННЫЕ ПЕРЕДАННЫЕ ПОЛЬЗОВАТЕЛЕМ,
    НУЖНО ПРОВЕРЯТЬ!
    $result = $model->show_last(intval($attributes['count']));
    // выводим шаблон
    $this->diafan->_tpl->get('show_last', 'example', $result);
}
```

example.model.php – модель

Пример:

```
public function show_last($count)
{
    if (! ($res = DB::query("SELECT text FROM {example} ORDER BY id DESC".(!
empty($count) ? ' LIMIT '.$count : "))))
        return false;

    $rows = array();
    while ($row = DB::fetch_array($res))
    {
        $rows[] = $row;
    }
    return $rows;
}
```

example.view.show.php – представление

Пример:

```
if (! $result)
{
    echo 'нет объявлений';
    return FALSE;
}

foreach ($result as $row)
{
    echo '<div class="example">'.$row['text'].'</div>';
}
```

Теперь в *themes/site.php*, либо на любой странице через визуальный редактор (из административной панели) пропишем:

```
<insert name="show_last" count="2" module="example">
```

И в этом месте мы увидим два последних объявления.

Но наш модуль все равно сыроват. При заходе на страницу с прикрепленным модулем мы ничего не видим. Давайте сделаем список объявлений, разбитый на страницы. При нажатии на объявление оно откроется в отдельном окне, в котором можно будет вписать комментарий.

Это не сложно, не пугайтесь :)

Пример:

```
public function show()
{

    // данные будем кэшировать
    $cache_meta = array(
        "name" => "list", // метка кэша
        "page" => $this->diafan->page > 1 ? $this->diafan->page : 1,
    );

    // если данных нет в кэше занесем их
    if (!$this->result = $this->diafan->_cache->get($cache_meta, 'example'))
    {
        $this->result = array();
    }

    ///navigation///
    $this->diafan->_paginator->page = $this->diafan->page;
    $this->diafan->_paginator->navlink = $this->diafan->_route->current_link("page");
    $this->diafan->_paginator->navlink_tpl = $this->diafan->_route->current_link("", array("page"
=> "%d"));
    $this->diafan->_paginator->nen = DB::query_result("SELECT COUNT(id) FROM {example}
WHERE act='1' AND trash='0'");
}
```

```

$this->result["paginator"] = $this->diafan->_paginator->get();
///navigation///

$result = DB::query_range("SELECT id, created, text FROM {example} WHERE act='1' AND
trash='0' ORDER BY created DESC, id DESC", $this->diafan->_paginator->polog, $this-
>diafan->_paginator->nastr);

while ($row = DB::fetch_array($result))
{
    $row['created'] = $this->format_date($row['created'], 'example');
    $row['link'] = $this->diafan->_route->link($this->diafan->cid, 'example', 0, $row['id']);

    $this->result['rows'][] = $row;
}
//сохранение кэша
$this->diafan->_cache->save($this->result, $cache_meta, 'example');
}

$this->result["paginator"] = $this->diafan->_tpl->get('get', 'paginator', $this-
>result["paginator"]);

return $this->result;
}

```

Как вы видите по коду здесь, мы использовали кэш. Это необходимо, если вы хотите уменьшить нагрузку на базу данных и сократить время отображения страницы.

На время разработки мы всегда рекомендуем вам отключать кэширование в параметрах сайта, потому что по умолчанию кэш сбрасывается лишь при изменении элемента в панели администрирования, то есть все ваши манипуляции с базой данных в файле **.model.php* со включенным кэшированием вы можете увидеть не сразу.

Накидаем небольшой пример, чтобы было понятнее

Пример:

```

$cache_meta = array(
    "name" => "trololo", // метка кеша
    "language" => _LANG,
    "page" => $this->diafan->page > 1 ? $this->diafan->page : 1,
    "site_id" => $this->diafan->cid,
);
// если кэш сохранен он будет в $data если нет то выполнится условие
$data = array();
if(! $data = $this->diafan->_cache->get($cache_meta, $this->diafan->module))
{
    $data = array('котейка', 'черный');
    //сохранение кэша
    $this->diafan->_cache->save($data, $cache_meta, $this->diafan->module);
}

```

Про постраничную навигацию [подробнее читайте здесь](#).

Остановимся подробнее на функции *format_date*. Эта функция наследуется из класса *model* и подробно описана по ссылке выше.

Она форматирует дату в соответствии с конфигурацией модуля.

Давайте добавим нашему модулю в конфигурацию кроме количества объявлений на странице еще и формат даты!

Создадим файл конфигурации *modules/example/admin/example.admin.config.php* и наполним его:

Пример:

```
public $variables = array (
    'base' => array (
        'nastr' => array(
            'type' => 'numtext',
            'name' => 'Количество объявлений на странице',
        ),
        'format_date' => array(
            'type' => 'select',
            'name' => 'Формат даты',
        ),
    )
);

public $select_arr = array(
    'format_date' => array(
        0 => '1.05.2009',
        1 => '1 мая 2008 г.',
        2 => '1 мая',
        3 => '1 мая 2008, понедельник',
        4 => 'не отображать',
    ),
);

public $config = array('config');
```

Получился функциональный модуль аж с двумя изменяемыми настройками и кэшированием. Для начала совсем не плохо, но сделаем еще лучше! Сделаем вывод отдельных объявлений с возможностью добавлять к ним комментарии.

(смотрите файл [example.module.comment.zip](#))

Модель и представление делаем по накатанной программе, а чтобы добавить комментарии, необходимо в конфиге (*example.admin.config.php*) подключить комментарии:

Пример:

```
public $variables = array(
    'main' => array(
```

```
'comments' => 'module',  
    ...  
),  
    ...  
);
```

и в модели вызвать обработчик

Пример:

```
if ($this->diafan->configmodules("comments"))  
{  
    $comments_html = $this->diafan->_comments->get();  
}
```

Функция *configmodules* проверяет, установлена ли данная переменная в конфигурации модуля, и, если да, подключает комментарии.

[Исходный код модуля example.](#)

Как добавить шаблонный тег

Шаблонные теги позволяют подключать исполняемый код в определенное место сайта.

Исполняемый код может быть общего характера или относиться к модулю.

Если нужно добавить **код общего характера**, то создаем общий шаблонный тег.

Для этого добавляем функцию в класс *Theme_functions*, описанный в файле *includes/function.php*. Новая функция должна иметь доступ *public*.

После этого шаблонный тег с названием функции можно использовать.

Функции можно передавать атрибуты тега. Для этого функция должна принимать аргумент – массив атрибутов.

Пример:

Добавляем шаблонную функцию *show_hello*.

//в файле includes/function.php Между любыми другими функциями в классе Theme_functions:

```
public function show_hello()
{
    echo "Hello";
}
```

Затем в любом основном шаблоне сайта, например в */themes/site.php*, функцию можно вызвать в любом месте шаблонным тегом

```
<insert name="show_hello">
```

Пример:

Добавляем модулю «Новости» шаблонную функцию *show_date*

//в файле modules/news/news.php

```
public function show_date($attributes)
{
    Customization::inc('modules/news/news.model.php');
    $model = new News_model($this->diafan);
    $result = $model->date($attributes["day"]);
    $this->diafan->_tpl->get('date', 'news', $result);
}
```

//в файле modules/news/news.model.php

```
public function date($day)
{
    if($day)
```

```
{  
    $date = $day.date(".m.Y");  
}  
else  
{  
    $date = date("d.m.Y");  
}  
return $date;  
}
```

```
//файл modules/news/views/news.view.date.php  
echo '<div class="news_date">'.$result.'</div>';
```

Затем в любом основном шаблоне сайта, например в *themes/site.php*, функцию можно вызвать в любом месте шаблонным тегом

```
<insert name="show_date" module="news">
```

Как добавить новое поле в модуль

При необходимости можно добавить в административную часть любого из модулей дополнительное поле и вывести его значение в пользовательской части.

В качестве примера добавим поле «Автор» в модуль новостей.

Для этого в таблицу `{news}` с помощью, например, [phpMyAdmin](#), добавим поле `author` с типом `varchar(250)`.

Выполним следующий SQL-запрос:

Пример:

```
ALTER TABLE `ВАШ_ПРЕФИКС_news` ADD `author` VARCHAR(250) NOT NULL;
```

Далее добавим в массив `$variables` файла `modules/news/admin/news.admin.php` в следующую запись:

Пример:

```
'main' => array (
    'author' => array(
        'type' => 'text',
        'name' => 'Автор'
    ),
    ...
)
```

где:

- `type` – тип поля (подробнее о типах полей можно посмотреть [здесь](#));
- `name` – название поля, которое будет отображаться в административной части.

Все! Этого достаточно, чтобы поле «Автор» появилось в административной части сайта на странице редактирования новости. Естественно, его можно заполнять, сохранять, редактировать.

Далее нужно вывести это поле где-нибудь в пользовательской части сайта, например, на странице списка новостей. Рассматривается случай, когда в настройках модуля **не подключены категории**.

Внесем изменения в файл `modules/news/news.model.php`. Корректировка затронет метод `list_query`, где происходит выборка всех полей новости.

Пример:

```
$result = DB::query_range("SELECT e.id, e.author, ...
```

Теперь, добавим в цикл вывода новостей файла `modules/news/views/news.view.list.php` код:

Пример:

```
if(! empty($row["author"]))
{
    echo $row["author"];
}
```

Все! У новостей в списке отобразится наше поле «Автор».

Если **категории подключены**, нужно править метод *list_category_elements_query* в файле *modules/news/news.model.php*:

Пример:

```
$result = DB::query_range("SELECT e.id, e.author, ...
```

Далее нужно действовать аналогично примеру с отключенными категориями. Файл *modules/news/views/news.view.list.php*:

Пример:

```
if(! empty($row["author"]))
{
    echo $row["author"];
}
```

Для вывода автора в **подробном описании новости** нужно внести изменения в метод *id_query* все того же файла *modules/news/news.model.php*:

Пример:

```
$row = DB::fetch_array(DB::query("SELECT id, e.author, ...
```

Пример:

```
$this->result["author"] = $row["author"];
```

Шаблон вывода подробного описания новости находится в файле *modules/news/views/news.view.id.php*. Имя автора хранится в *\$result['author']*. Чтобы его отобразить добавим в нужное место файла код:

Пример:

```
if(! empty($result['author']))
{
    echo $result['author'];
}
```

Возможно, понадобится вывести имя автора в **блоке последних новостей**. Откроем файл *modules/news/news.model.php* и в метод *show_block()* внесем следующие изменения:

Пример:

```
$result = DB::query_range("SELECT e.id, e.author, ...
```

Далее в файле *modules/news/views/news.view.show_block.php* в цикле вывода новостей добавим:

Пример:

```
if(! empty($row["author"]))  
{  
    echo $row["author"];  
}
```

Добавление полей в другие модули осуществляется аналогично приведенным выше примерам.

Как получить свободные данные из БД и вывести их

На данном примере рассмотрим создание шаблонного тега вывода 5 последних комментариев и аватара пользователя, добавившего комментарий.

Исходные коды всех шаблонных тегов, не относящихся к какому-либо модулю, находятся в файле *includes/function.php*. В этом файле единственный класс *Theme_functions*, который и содержит все методы. Добавим наш метод для шаблонного тега. Назовем его *show_last_comments*:

Пример:

```
public function show_last_comments()
{
    echo 'diafan.CMS';
}
```

Теперь, если в одном из шаблонов, расположенных в папке *themes*, написать шаблонный тег `<insert name="show_last_comments">`, то на сайте мы увидим надпись «diafan.CMS».

Расширим наш метод, а именно – выберем с помощью SQL-запроса последние 5 активных комментариев, а также имена пользователей, их добавивших.

Пример:

```
DB::query("SELECT u.name, c.text FROM {comments} AS c LEFT JOIN {users} AS u ON c.user_id=u.id WHERE c.act='1' AND c.trash='0' ORDER BY c.id DESC LIMIT 5");
```

Внимание! LEFT JOIN в данном случае используем для того, чтобы выбрать все последние комментарии независимо от того, кто их оставил – авторизованный пользователь или гость. Для не авторизованных пользователей поле *user_id* таблицы *{comments}* имеет значение 0. Для комментариев, оставленных авторизованным пользователем, это поле соответствует *id* из таблицы *{users}*.

Теперь метод примет такой вид:

Пример:

```
public function show_last_comments()
{
    $result = DB::query("SELECT u.name, c.text FROM {comments} AS c LEFT JOIN {users} AS u ON c.user_id=u.id WHERE c.act='1' AND c.trash='0' ORDER BY c.id DESC LIMIT 5");
}
```

Но в таком виде он ничего не выводит. Давайте выведем текст комментариев:

Пример:

```

$text = "";
while ($row = DB::fetch_array($result))
{
    $text .= $row['text'].'<br>';
}
echo $text;

```

Исходный код стал выглядеть так:

Пример:

```

public function show_last_comments()
{
    $result = DB::query("SELECT u.name, c.text FROM {comments} AS c LEFT JOIN {users} AS u ON c.user_id=u.id WHERE c.act='1' AND c.trash='0' ORDER BY c.id DESC LIMIT 5");
    $text = "";
    while ($row = DB::fetch_array($result))
    {
        $text .= $row['text'].'<br>';
    }
    echo $text;
}

```

Имея в наличии имена пользователей, вывести их аватары не составит труда. Но перед выводом необходимо проверить, включена ли в настройках системы возможность их добавления:

Пример:

```

if ($this->diafan->configmodules("avatar", "users"))

```

Если возможность есть, и существует изображение аватара, то можно вывести:

Пример:

```

if ($this->diafan->configmodules("avatar", "users") &&
file_exists(ABSOLUTE_PATH.USERFILES.'/avatar/'.$row['name'].'.png'))
{
    $avatar = BASE_PATH.USERFILES.'/avatar/'.$row['name'].'.png';
    $avatar_width = $this->diafan->configmodules("avatar_width", "users");
    $avatar_height = $this->diafan->configmodules("avatar_height", "users");
    $text .= '<br>';
}

```

Получаем итоговый код:

Пример:

```

public function show_last_comments()

```

```

{
    $result = DB::query("SELECT u.name, c.text FROM {comments} AS c LEFT JOIN {users} AS
u ON c.user_id=u.id WHERE c.act='1' AND c.trash='0' ORDER BY c.id DESC LIMIT 5");
    $text = "";
    while ($row = DB::fetch_array($result))
    {
        if ($this->diafan->configmodules("avatar", "users") &&
file_exists(ABSOLUTE_PATH.USERFILES.'/avatar/'.$row['name'].'.png'))
        {
            $avatar = BASE_PATH.USERFILES.'/avatar/'.$row['name'].'.png';
            $avatar_width = $this->diafan->configmodules("avatar_width", "users");
            $avatar_height = $this->diafan->configmodules("avatar_height", "users");
            $text .= '<br>';
        }
        $text .= $row['text'].'<br>';
    }
    echo $text;
}

```

Затем в любом основном шаблоне сайта, например в *themes/site.php*, функцию можно вызвать в любом месте шаблонным тегом `<insert name="show_last_comments">`.

Как вставить сторонний PHP-код

Вставку и выполнение стороннего кода можно осуществить с помощью создания собственного шаблонного тега.

Для этого в *includes/function.php* добавим еще один метод. Например:

Пример:

```
public function show_my_code()
{
    echo 'diafan.CMS';
}
```

Чтобы код внутри метода выполнялся, выведем шаблонный тег в одном из файлов папки *themes*. Как правило, это *site.php*:

```
<insert name="show_my_code">
```

Теперь код метода *show_my_code()* будет выполняться каждый раз, когда шаблоном страниц сайта является файл *site.php*. В нашем случае будет выводиться «diafan.CMS».

Тег также можно вставить в визуальный редактор, для этого нужно выбрать кнопку «Редактировать HTML-код» (первая кнопка визуального редактора).

Как сделать зависимые Ajax-списки

Данный пример покажет возможность создания зависимых выпадающих списков в форме поиска каталога.

Внимание! Пример применим для небольшого массива данных. Если планируется большой список зависимых значений (скажем, больше 100), то лучше применить другое решение.

Рассматривать пример будем с использованием установленных демо-данных, а именно на примере «Бумажного зоопарка».

Предположим, что нам понадобилось разделить животных по странам обитания. Т.к. стран большое количество, логичным было бы их как-то сгруппировать, чтобы облегчить конечному пользователю поиск нужной страны. Группировкой в данном случае будут служить материки.

Добавим в каталоге новую характеристику с типом «выпадающий список» и внесем данные. В результате получаем следующее:

The screenshot shows a web interface for adding a new characteristic to a catalog. The form is divided into several sections:

- Header:** "Название" (Name) with the value "Материк" (Continent) and a question mark icon.
- Type:** "Тип" (Type) with a dropdown menu set to "выпадающий список" (dropdown list) and a question mark icon.
- Values:** "Значения, псевдоссылка" (Values, pseudolink) with a list of six items: "Евразия", "Северная Америка", "Южная Америка", "Африка", "Австралия", and "Антарктида". Each item has an adjacent empty input field.
- Separate Page:** "Отдельная страница для значений" (Separate page for values) with an unchecked checkbox.
- Category:** "Категория" (Category) with a dropdown menu showing "Птицы", "Звери", "Крупные", and "Мелкие".
- Sorting:** "Сортировка: установить перед" (Sorting: set before) with a link to "Материк".
- Display Options:** Three checkboxes: "Использовать в форме поиска" (checked), "Показывать в списке", and "Показывать в блоке товаров".
- Description:** "Описание характеристики" (Characteristic description) with a large text area.

Добавим аналогичную характеристику для стран.

Зависимость характеристик будем хранить в дополнительном поле таблицы `{shop_param_select}`, которая содержит все значения выпадающих списков. Подробнее о структуре таблицы можно узнать в разделе [Структура базы данных diafan.CMS](#).

Выполним следующий SQL-запрос, например в phpMyAdmin:

Пример:

```
ALTER TABLE `shop_param_select` ADD `parent_param_id` INT(11) NOT NULL
```

Теперь откроем файл `modules/shop/admin/shop.admin.param.php`, который отвечает за вывод подробного описания характеристики в административной части.

Понадобится нам метод `edit_variable_param_select()`, отвечающий за редактирование характеристики.

Найдем в методе следующие строки:

```
if (in_array($this->diafan->values["type"], array('select', 'multiple')))  
{
```

и после них добавим:

Пример:

```
if($this->diafan->edit == 12)  
{  
    $mainland = array();  
    $result = DB::query("SELECT id, [name] FROM {shop_param_select} WHERE param_id=  
%d", 11);  
    while($row = DB::fetch_array($result))  
    {  
        $mainland[$row['id']] = $row['name'];  
    }  
}
```

Здесь мы проверяем, что в данный момент открыта нужная характеристика. Нужная в нашем случае «Страна» с ID, равным 12.

Внимание! В вашем случае ID может иметь любое другое значение. Чтобы узнать ID характеристики, достаточно навести на нее курсор мыши в общем списке характеристик в административной части.

Далее из таблицы `{shop_param_select}` выбираются все значения характеристики, они же значения выпадающего списка, с ID равным 11, т.е. материи. Полученные данные записываются в массив `$mainland` в виде `id_значения => имя_значения`.

Теперь выведем данный массив в виде выпадающего списка.

В этом же методе, в цикле, начинающемся с

```
foreach ($value as $row)
{
```

после строки

```
<input type="text" name="param_rewrite[]" size="20" value="" . $row["rewrite"] . "">
```

напишем

Пример:

```
;;
if($this->diafan->edit == 12)
{
    $selected = DB::query_result("SELECT parent_param_id FROM {shop_param_select}
WHERE id=%d AND param_id=%d", $row["id"], 12);
    echo '<select name="mainland'.$row["id"].">';
    foreach($mainland as $key=>$value)
    {
        echo '<option value="'.$key.'" '
        .($selected == $key ? 'selected' : '')
        .'>'.$value.'</option>';
    }
    echo '</select>';
}
echo '
```

Здесь мы опять проверяем, что находимся в нужной характеристике, а далее циклом формируем выпадающий список на основе полученного ранее массива `$mainland`.

Обратите внимание на имя списка `"mainland".$row["id"]`. Здесь `$row["id"]` — это значение ID текущего элемента, выпадающего списка «Страна».

При этом в каждой итерации делаем сравнение значения нового поля `parent_param_id`, которое хранится в `$selected`, с `$key`. В `$key` находится ID элемента выпадающего списка «Материки». Если значения совпали, то добавляем к текущему пункту атрибут `selected`, т.е. делаем его выбранным.

Теперь характеристика «Страна» примет такой вид.

Название Страна ?

Тип выпадающий список ?

Значения, псевдоссылка

Россия		Евразия
США		Северная Америка
Великобритания		Евразия
Франция		Евразия
Китай		Евразия
Индия		Евразия
Пакистан		Евразия
КНДР		Евразия
Израиль		Евразия

Отдельная страница для значений ☐

Категория

Птицы
Звери
Крупные
Мелкие

Сортировка: установить перед Страна

Использовать в форме поиска ☒ ?

Показывать в списке ☐

Показывать в блоке товаров ☐

Описание характеристики

Мы добились вывода списка материков, но не сделали их сохранение. За сохранение характеристик отвечает метод `save_variable_param_select()`. Немного его модифицируем. Перед

```
$ids[] = $id;
```

добавим

Пример:

```
if(!empty($_POST['mainland'].$id))
{
    DB::query("UPDATE {shop_param_select} SET parent_param_id=%d WHERE id=%d AND
param_id=%d", $_POST['mainland'].$id, $id, 12);
}
```

В данном коде нет ничего сложного — проверяется наличие переменной `mainland` и выполняется обновление нового поля `parent_param_id`.

Административная часть теперь вполне функциональна и удобна.

Перейдем к пользовательской части.

Для начала выведем характеристики «Материк» и «Страна» в форме поиска в виде

выпадающего списка.

Внимание! По умолчанию все выпадающие списки выводятся в виде чекбоксов, чтобы дать пользователю возможность выбора сразу нескольких значений.

Для этого в файле *modules/shop/views/shop.view.show_search.php* код

Пример:

```
case 'select':
case 'multiple':
echo '<span class="infofield">'.$row["name"].':</span>';
foreach ($row["select_array"] as $key => $value)
{
    echo '<input type="checkbox" name="p'.$row["id"].'[]" value="'.$key.'"'.(in_array($key,
$row["value"]) ? " checked" : "").' class="inpcheckbox"> '.$value.'<br>';
}
```

заменим на

Пример:

```
case 'select':
case 'multiple':
if($row["id"] == 11 || $row["id"] == 12)
{
    echo '<span '.($row["id"] == 12 ? 'style="display: none;" class="country" :
'class="mainland"'>'.>
    <span class="infofield">'.$row["name"].':</span>
    <select name="p'.$row["id"].'[]">';
    echo '<option value="0"></option>';
    foreach ($row["select_array"] as $key => $value)
    {
        echo '<option value="'.$key.'"'.(in_array($key, $row["value"]) ? " selected" : "").'>'.
$value.'</option>';
    }
    echo '</select></span>';
}
else
{
    echo '<span class="infofield">'.$row["name"].':</span>';
    foreach ($row["select_array"] as $key => $value)
    {
        echo '<input type="checkbox" name="p'.$row["id"].'[]" value="'.$key.'"'.(in_array($key,
$row["value"]) ? " checked" : "").' class="inpcheckbox"> '.$value.'<br>';
    }
}
```

В данной части кода проверяются ID характеристик. Если это нужные нам 11 или 12, формируем выпадающий список, иначе выводим характеристику по умолчанию, т.е. чекбоксами.

При этом характеристику «Страна» делаем скрытой.

Следующим шагом будет создание метода отправки POST запроса при выборе значения характеристики «Материк».

Откроем файл *js/user-func.js* и после

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

добавим

Пример:

```
$(".mainland select").change(function(){
    var selected = $(".mainland select option:selected").val();
    $(".country select option").remove();
    $.ajax({
        type: "POST",
        url: "",
        data: "module=shop&action=change_country&mainland=" + selected,
        success: function (response, textStatus)
        {
            $(".country select").append( $(response) );
            $(".country").show();
        }
    });
});
```

Про метод *ajax* библиотеки *jQuery* написано очень много статей, поэтому найти его описание не составит труда. Мы же остановимся на переменных, которые передаются системе управления.

module=shop — сообщает, что обращаемся мы к файлу *modules/shop/shop.ajax.php*.

action=change_country — здесь в произвольной форме указываем имя метода, который будет обрабатывать запрос (создадим мы его чуть позже).

mainland= + selected — текущее значение выпадающего списка.

Перейдем к написанию обработчика.

Как было сказано выше, обработчик нам необходимо делать в файле *modules/shop/shop.ajax.php*.

В метод *ajax_request()* после

```
switch ($_POST['action'])
{
```

добавим

Пример:

```
case 'change_country':  
return $this->change_country();
```

И добавим новый метод:

```
private function change_country()  
{  
    $text = '<option value="0">-</option>';  
    $result = DB::query("SELECT id, [name] FROM {shop_param_select} WHERE  
parent_param_id=%d", $_POST['mainland']);  
    while($row = DB::fetch_array($result))  
    {  
        $text .= '<option value="'. $row['id']. ">'. $row['name']. '</option>';  
    }  
  
    echo $text;  
    return TRUE;  
}
```

В данном методе происходит выборка элементов выпадающего списка на основе полученного значения ID материка, которое хранится в `$_POST['mainland']`.

Далее формируется выпадающий список и происходит его вывод.

Поиск по товарам

Название:

Артикул:

Описание:

Цена:

от до

Содержание:

- ☐ Домашнее
- ☐ Во дворе
- ☐ Дикое

Цвет:

- ☐ желтый
- ☐ голубой
- ☐ фиолетовый
- ☐ оранжевый
- ☐ зеленый
- ☐ черный
- ☐ розовый
- ☐ коричневый

Материк:

Северная Америка ▼

Страна:

-

▼

-

США

К

Как в модуле использовать другой модуль

На данном примере мы рассмотрим возможность прикрепления одного из фото модуля фотогалереи к отдельной статье.

Для начала добавим дополнительное поле в таблицу `{clauses}`, в котором будем хранить идентификатор прикрепленной фотографии. Выполним следующий SQL-запрос, например в phpMyAdmin:

Пример:

```
ALTER TABLE `clauses` ADD `photo` INT(11) NOT NULL
```

Теперь внесем изменения в файл `modules/clauses/admin/clauses.admin.php`.

Добавим наше новое поле в массив `$variables`:

Пример:

```
public $variables = array (  
    'main' => array (  
        'photo' => array(  
            'type' => 'function',  
            'name' => 'Фото из фотогалереи'  
        ),  
        ...  
    ),  
    ...  
);
```

Тип поля указываем `function` – это позволит определить методы, в которых мы сможем самостоятельно описывать действия, выполняемые при выводе и сохранении нового поля.

Теперь добавим первый метод, который будет выполняться всякий раз, когда мы заходим на страницу редактирования статьи в административной части:

Пример:

```
public function edit_variable_photo()  
{  
    echo '  
<tr valign="top">  
    <td class="td_first">Фото из фотогалереи</td>  
    <td>';  
  
    $result = DB::query("SELECT id, [name] FROM {photo} WHERE [act]='1' AND trash='0'");  
  
    echo '<select name="photo">';  
    echo '<option value="0">-</option>';  
    while ($row = DB::fetch_array($result))  
    {  
        echo '<option value="'. $row['id']. "' . ($this->diafan->values['photo'] == $row['id'] ?  
'selected' : '').>'. $row['name'].</option>';  
    }  
}
```

```

    }
    echo '</select>'
  }
}

```

Рассмотрим некоторые ключевые моменты при создании подобных методов.

Во-первых, это его именование. Оно обязательно должно быть вида `edit_variable_имя_поля`.

Во-вторых, это цикл, в котором выводятся все элементы в виде выпадающего списка. В нем мы сравниваем значение идентификатора фотографии и текущее значение поля `photo`. Все текущие значения поля хранятся в массиве `$this->diafan->values`. Для нас важно значение элемента `$this->diafan->values['photo']`, чтобы добавить атрибут «selected» пункту выпадающего списка, если значение поля и идентификатора совпадают.

Результат можно увидеть на скриншоте:

Страницы сайта ▸ Статьи

Редактировать

Фото из фотогалереи: Жираф Инокентий ▾

Название: Коршун Маркус

Дата: Лошадь Ксения

Показывать на сайте: Козел Вова

Изображения: Слон Антон

Похожие статьи: Крокодил Геннадий

Анонс?: Жираф Инокентий

Корова Варя

Страус Ахмед

Кобра Жанна

Цапля Денис

Черепаха Коля

Заяц Кирилл

Рыба Валентина

Утенок Магкряк

Орел Егорка

Мышь Пика

Лиса Лия

Гриф Вазген

Суслик Миша

Обзор

Второй метод, который нам необходим, выглядит так:

Пример:

```

public function save_variable_photo()
{
    if(!empty($_POST['photo']))
    {

```

```

        DB::query("UPDATE {clauses} SET photo=%d WHERE id=%d", $_POST['photo'], $this->diafan->save);
    }
}

```

Здесь также важно наименование – оно аналогично первому методу, только вместо *edit* пишем *save*. Метод срабатывает при сохранении элемента, в нашем случае отдельной статьи.

Суть его предельна проста – сохраняем выбранное пользователем значение выпадающего списка фотографий в поле *photo* для текущей статьи.

Далее выведем фотографию на подробной странице статьи.

Открываем файл *modules/clauses/clauses.model.php* и вносим изменения в метод *id_query*:

Пример:

```
$row = DB::fetch_array(DB::query("SELECT id, photo, ...
```

а также добавляем в метод *id* следующий код:

Пример:

```

$photo_array = DB::fetch_array(DB::query("SELECT [name], site_id FROM {photo} WHERE id=%d", $row["photo"]));

$photo = $this->diafan->_images->get(
    'large', $row["photo"], 'photo', $photo_array['site_id'], $photo_array['name']
);

$this->result["photo"] = $photo[0];

```

где *\$photo_array* – массив, содержащий значения *site_id* и имена, необходимые для метода *\$this->diafan->_images->get* (подробнее о методе и о возвращаемых им значениях можно посмотреть [здесь](#)).

Теперь в *\$this->result["photo"]* хранится значение ширины, высоты пути к файлу изображения и т.д.

Откорректируем шаблон страницы.

Для этого в файл *modules/clauses/views/clauses.view.id.php* добавим строки:

Пример:

```

if(!empty($result['photo']))
{
    echo '<img src='.$result['photo']['src'].' width="'.$result['photo']['width'].'" height="'.$result['photo']['height'].'" alt="'.$result['photo']['alt'].'" title="'.$result['photo']['title'].'">';
}

```

}

Результат на следующем скриншоте:

Головоногий пингвин-бобёр



15 октября

Самка слона и ее друг крокодил родили на свет нового обитателя нашего зоопарка - головоногого пингвина-б

Стоит рассказать, как произошла роковая встреча Слонихи и Крокодила. Была черная ночь. Слонихе, как и в услышала дивную мелодию. Это пел крокодил. Он чистил зубы маленькой птичкой перед сном. Слонихе оче их чувств. Мастер долго веретел лист бумаги и, наконец, принялся за работу.

В итоге получился зверек, которому впоследствии дали имя Крокослон и отнесли к виду Головоногого пингви

Теги: [Пингвин](#) , [Бобер](#) , [Фиолетовый](#)



Переопределение функционала в административной части модуля

Приведенный ниже пример покажет, как реализовать отправку уведомления на почтовый ящик администратора при сохранении новости.

Откроем файл *modules/news/admin/news.admin.php* и добавим в массив *\$variables* новый элемент с типом «none»:

Пример:

```
'main' => array (  
    'sendmail' => array(  
        'type' => 'none',  
    ),  
    ...  
);
```

Тип «none» позволяет не выводить поле в административной части, но при этом выполнять необходимый код при сохранении и редактировании элемента, в нашем случае новости (подробнее о типах полей можно посмотреть [здесь](#)).

Также добавим дополнительное поле в таблицу *{news}*. Выполним следующий SQL-запрос, например в phpMyAdmin:

Пример:

```
ALTER TABLE `news` ADD `sendmail` INT(11) NOT NULL
```

В данном случае тип поля не важен, т.к. хранить в нем мы ничего не будем.

Далее в этот же файл добавим метод:

Пример:

```
public function save_variable_sendmail()  
{  
  
}
```

Он будет выполняться при каждом сохранении новости.

Теперь перейдем к реализации требуемого функционала.

Для отправки почты в diafan.CMS используется функция *send_mail()*. Для ее выполнения необходимо подключить файл *includes/mail.php*. Подробное описание функции можно найти в соответствующем [разделе](#).

Обязательными для функции являются три параметра — адрес получателя, тема письма и текст письма.

Адресом получателя в нашем случае будут являться почтовые адреса администраторов сайта. Данные о пользователях хранятся в таблице `{users}`. Все администраторы сайта в поле `role_id` имеют значение 3. Зная это, мы можем выбрать нужные нам почтовые ящики:

Пример:

```
$result = DB::query("SELECT DISTINCT(id), mail FROM {users} WHERE role_id=%d", 3);
```

Тему письма можем указать произвольную, например так:

Пример:

```
$subject = 'Изменения на сайте '.TIT1;
```

TIT1 это константа, содержащая имя сайта, указанное в разделе «Параметры сайта» административной части.

В тексте письма будем отправлять ссылку на новость, которая претерпела изменения.

Для формирования ссылок используется метод `$this->diafan->_route->link()`. Подробно метод описан в разделе [Маршрутизатор](#).

Для использования метода нам понадобятся значения ID страницы, к которой прикреплен модуль, название модуля, ID категории, если категории подключены в настройках модуля и ID текущей новости.

ID страницы хранится в переменной `$this->diafan->site`.

Модуль, в нашем случае, имеет название «news».

ID категории будем проверять наличием переменной `$_POST['cat_id']`. Если ее нет, то в качестве значения будем использовать 0.

ID новости хранится в `$this->diafan->save`.

В итоге получим такой код:

Пример:

```
$link = $this->diafan->_route->link(  
    $this->diafan->site,  
    'news',  
    (!empty($_POST['cat_id']) ? $_POST['cat_id'] : 0),  
    $this->diafan->save  
);
```

Теперь остается циклом перебрать все почтовые ящики администраторов и отправить им уведомления. Результат будет выглядеть так:

Пример:

```

public function save_variable_sendmail()
{
    //выбираем значения почтовых ящиков администраторов
    $result = DB::query("SELECT DISTINCT(id), mail FROM {users} WHERE role_id=%d", 3);

    //тема письма
    $subject = 'Изменения на сайте '.TIT1;

    //ссылка на текущую новость
    $link = $this->diafan->_route->link(
        $this->diafan->site,
        'news',
        (!empty($_POST['cat_id']) ? $_POST['cat_id'] : 0),
        $this->diafan->save
    );

    //текст сообщения
    $message = '<a href="'.BASE_PATH . $link .'">'.(!empty($_POST['name']) ? $_POST['name']
: $this->diafan->save).'</a>';

    //делаем рассылку
    include_once(ABSOLUTE_PATH."includes/mail.php");
    while ($row = DB::fetch_array($result))
    {
        send_mail(
            $row["mail"],
            $subject,
            $message
        );
    }
}

```


Кастомизация функций

Кастомизация позволяет дорабатывать файлы diafan.CMS так, чтобы при автообновлении внесенные изменения не были перезаписаны. Для этого все доработки нужно размещать в файле с названием файла оригинала, но суффиксом `.custom`. Общие принципы работы данного механизма приведены в разделе [Кастомизация](#).

Мы рассмотрим пример, взяв за основу файл `includes/core.php`.

При добавлении изображений в новостях, статьях и т.д. имена файлов формируются на основе имени элемента, в котором происходит сохранение. При этом они автоматически преобразуются в латинские символы с помощью метода `translit()`. Пробел по умолчанию заменяется на дефис. Предположим, что перед нами встала необходимость менять пробел на символ нижнего подчеркивания и приводить все символы строчному виду.

Создадим новый файл `includes/core.custom.php` и добавим в него новые методы.

Пример:

```
abstract class Core
{
    replace public function translit ($text)
    {
        $ru = array('а', 'б', 'в', 'г', 'д', 'е', 'ё', 'ж', 'з', 'и', 'й', 'к', 'л', 'м', 'н', 'о', 'п', 'р', 'с', 'т', 'у', 'ф', 'х',
        'ц', 'ч', 'ш', 'щ', 'ы', 'э', 'ю', 'я', 'А', 'Б', 'В', 'Г', 'Д', 'Е', 'Ё', 'Ж', 'З', 'И', 'Й', 'К', 'Л', 'М', 'Н', 'О', 'П', 'Р',
        'С', 'Т', 'У', 'Ф', 'Х', 'Ц', 'Ч', 'Ш', 'Щ', 'Ы', 'Э', 'Ю', 'Я', ' ');
        $tr = array('a', 'b', 'v', 'g', 'd', 'e', 'yo', 'zh', 'z', 'i', 'y', 'k', 'l', 'm', 'n', 'o', 'p', 'r', 's', 't', 'u', 'f', 'kh',
        'ts', 'ch', 'sh', 'sch', 'y', 'e', 'yu', 'ya', 'A', 'B', 'V', 'G', 'D', 'E', 'YO', 'ZH', 'Z', 'I', 'Y', 'K', 'L', 'M', 'N', 'O',
        'P', 'R', 'S', 'T', 'U', 'F', 'KH', 'TS', 'CH', 'SH', 'SCH', 'Y', 'E', 'YU', 'YA', '_');
        $text = str_replace($ru, $tr, $text);
        $text = $this->prepare_translit($text);
        return $text;
    }
    new private function prepare_translit($text)
    {
        $text = strtolower($text);
        return $text;
    }
}
```

Здесь следует обратить внимание на слово `replace` перед модификатором доступа `public` метода `translit()`. Оно говорит системе, что данный метод уже существует в исходном файле, но использовать его не нужно. т. е. всякий раз при обращении к оригинальному `translit()` будет срабатывать `translit()` из файла `includes/core.custom.php`.

`New` позволяет добавить новый метод в существующий класс, что мы и сделали на примере `prepare_translit()`.

Помимо новых методов можно создавать и новые свойства класса. Сделаем это с переменными `$ru` и `$tr`.

Пример:

```
abstract class Core
{
    new private $ru = array('а', 'б', 'в', 'г', 'д', 'е', 'ё', 'ж', 'з', 'и', 'й', 'к', 'л', 'м', 'н', 'о', 'п', 'р', 'с', 'т',
        'у', 'ф', 'х', 'ц', 'ч', 'ш', 'щ', 'ы', 'э', 'ю', 'я', 'А', 'Б', 'В', 'Г', 'Д', 'Е', 'Ё', 'Ж', 'З', 'И', 'Й', 'К', 'Л', 'М', 'Н',
        'О', 'П', 'Р', 'С', 'Т', 'У', 'Ф', 'Х', 'Ц', 'Ч', 'Ш', 'Щ', 'Ы', 'Э', 'Ю', 'Я', ' ');

    new private $tr = array('a', 'b', 'v', 'g', 'd', 'e', 'yo', 'zh', 'z', 'i', 'y', 'k', 'l', 'm', 'n', 'o', 'p', 'r', 's', 't', 'u',
        'f', 'kh', 'ts', 'ch', 'sh', 'sch', 'y', 'e', 'yu', 'ya', 'A', 'B', 'V', 'G', 'D', 'E', 'YO', 'ZH', 'Z', 'I', 'Y', 'K', 'L', 'M',
        'N', 'O', 'P', 'R', 'S', 'T', 'U', 'F', 'KH', 'TS', 'CH', 'SH', 'SCH', 'Y', 'E', 'YU', 'YA', '_');

    replace public function translit ($text)
    {
        $ru = $this->ru;
        $tr = $this->tr;
        $text = str_replace($ru, $tr, $text);
        $text = $this->prepare_translit($text);
        return $text;
    }
    new private function prepare_translit($text)
    {
        $text = strtolower($text);
        return $text;
    }
}
```

Помимо *new* и *replace* существуют также *before* и *after*, позволяющие расширить метод и добавить необходимый код в его начало или конец, соответственно. Используются они аналогично примерам, приведенным выше.

Изменение административной части

Иногда требуется сделать административный интерфейс более информативным. Например, отобразить дополнительную информацию в каком-то списке (новостей, товаров и т.д.), чтобы не заходить в каждый элемент в отдельности.

В данном примере рассмотрим возможность добавления отображения способа оплаты в списке заказов.

За формирование списка заказов отвечает файл `modules/shop/admin/shop.admin.order.php`.

Перечень данных, которые нужно отображать в списке, находится в массиве `$config_other_row`.

Добавим в него новый элемент `'payment_id' => 'function'`. В результате массив примет такой вид:

Пример:

```
public $config_other_row = array (  
    'status_id' => 'function',  
    'summ' => 'function',  
    'payment_id' => 'function',  
    'user_id' => 'function',  
);
```

Для каждого из элементов массива определен метод, который формирует выводимые данные. Добавим его и для нового элемента:

Пример:

```
public function other_row_payment_id($row)  
{  
    $payment = DB::query_result("SELECT [name] FROM {shop_payment} WHERE id=%d",  
    $row['payment_id']);  
    return "</td><td class='payment_id' width='100px'>".$payment;  
}
```

Рассмотрим метод подробнее.

Обязательно следует обратить внимание на именование метода. Он всегда начинается с `other_row_`, а далее идет значение ключа элемента массива `$config_other_row`. В нашем случае это `payment_id`.

В метод передается массив `$row`, который содержит ряд параметров текущего заказа, в том числе нужный нам `$row['payment_id']`.

`$row['payment_id']` — это ID способа оплаты таблицы `{shop_payment}`. Он позволяет нам выбрать имя способа оплаты для конкретного заказа. Сохраняем имя в переменной `$payment`.

Далее мы просто выводим его в общем списке.

Данный пример универсален для всех модулей. При необходимости вы можете самостоятельно вывести дополнительные данные в любом из списков.

Как в поиске по товарам заменить чекбоксы на список

Вопрос очень простой, но регулярно поступающий в службу поддержки.

В шаблоне поиска по товарам *modules/shop/views/shop.view.show_search.php* находим цикл по дополнительным характеристикам:

Пример:

```
foreach ($result["rows"] as $row)
{
    //...
}
```

Характеристики с типами «список» и «список с выбором нескольких значений» выводятся следующими строками:

Пример:

```
case 'select':
case 'multiple':
    echo '
    <span class="infofield">'.$row["name"].':</span>';
    foreach ($row["select_array"] as $key => $value)
    {
        echo '<input type="checkbox" name="p'.$row["id"].'["' value="'.$key.'"'.(in_array($key,
$row["value"]) ? " checked" : "").' class="inpcheckbox"> '.$value.'<br>';
    }
}
```

Заменяем на следующий код:

Пример:

```
case 'select':
case 'multiple':
    echo '
    <span class="infofield">'.$row["name"].':</span>
    <select name="p'.$row["id"].'[">';
    foreach ($row["select_array"] as $key => $value)
    {
        echo '<option value="'.$key.'"'.(in_array($key, $row["value"]) ? " selected" : "").'>'.
$value.'</option>';
    }
    echo '</select>';
```

Пример можно применять и в модуле «Объявления».

Как вывести дополнительную характеристику товара в любом месте

Дополнительные характеристики товара выводятся циклом в порядке сортировки в административной части. Для них даже есть отдельный шаблон *modules/shop/views/shop.view.param.php*.

Но бывает так, что нужно какую-то отдельную характеристику вывести в определенном месте. Например, есть характеристика «Цвет» и нам информацию о цвете нужно вывести в таком виде:

Пример:

Цвет:



Есть два способа выбрать характеристику: по идентификатору и по имени.

Массив характеристик с идентификатором по имени – `$result["names_param"]`. В этом случае код будет такой:

Пример:

```
if(! empty($result["names_param"]["Цвет"]))
{
    echo 'Цвет: ';
    foreach($result["names_param"]["Цвет"]["value"] as $value)
    {
        echo '';
    }
}
```

В примере мы выводили характеристику с типом «список с выбором нескольких значений». Для нее задается несколько значений, поэтому переменная `$result["names_param"]["Цвет"]` задана как массив. Все остальные переменные имеют единственное значение.

Например, для текстовой характеристики «Материал» код будет следующим:

Пример:

```
if(! empty($result["names_param"]["Материал"]))
{
    echo 'Материал: '.$result["names_param"]["Материал"]["value"];
}
```

Если нужно использовать идентификатор вместо имени, тогда обращаемся к массиву `$result["ids_param"]`.

Пример:

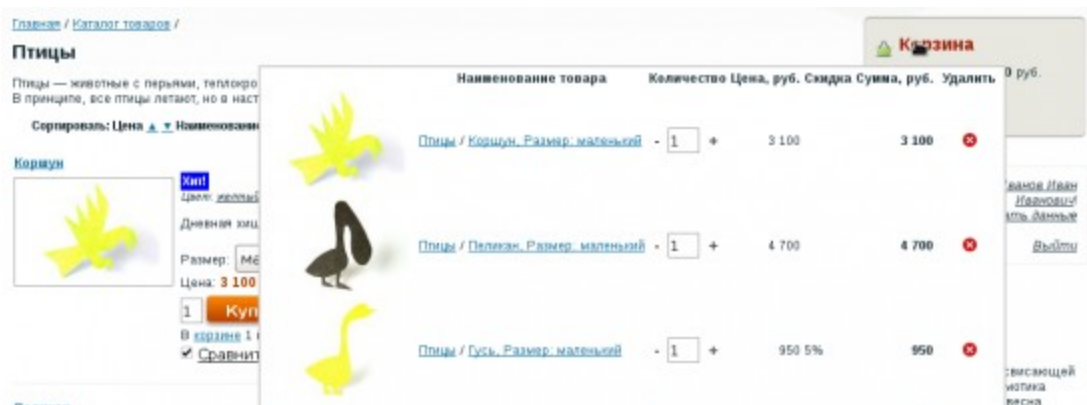
```
if(! empty($result["ids_param"][3]))
{
    echo $result["ids_param"][3]["name"].': '.$result["ids_param"][3]["value"];
}
```

Выше рассмотрены случаи, когда характеристику выводим в карточке товара. В остальных случаях заменяем `$result` на `$row`. Чтобы вывести характеристику в блоке или списке, для нее обязательно должна быть отмечена соответствующая опция.

Пример можно применять и в модуле «Объявления».

Как вывести список товаров в корзине в шаблонном теге

Бывает недостаточно просто надписи «Корзина: товаров 3 на 3 450 руб.». Нужно вывести всю корзину. Например:



Добавим таблицу товаров в шаблонный тег `show_block module="cart"`. Для этого в начале функции `show_block()` в файле `modules/cart/cart.model.php` добавим код:

Пример:

```
public function show_block()
{
    $result = $this->form_table();
    //...
}
```

И в шаблоне `modules/cart/views/cart.view.info.php` добавим:

Пример:

```
if(! empty($result["rows"]))
{
    echo $this->get('table', 'cart', $result);
}
```

Теперь ниже надписи «Корзина: товаров 3 на 3 450 руб.» будет выводиться таблица товаров. Остальное – дело верстки.

Если нужно, чтобы таблица товаров выглядела по-другому, не так, как на странице корзины – скопируйте файл `modules/cart/views/cart.view.table.php`. Назовите его, например, `modules/cart/views/cart.view.block_table.php`. И в шаблоне `modules/cart/views/cart.view.info.php` подключите этот файл:

Пример:

```
echo $this->get('block_table', 'cart', $result);
```

В новом файле настройте вид таблицы с товарами так, как нужно.

Чтобы в блоке корзины работали кнопки «Пересчитать», «Удалить» и др., нужно:

1. Добавить следующий код в шаблон *modules/cart/views/cart.view.info.php*

Пример:

```
if(! empty($result["rows"]))
{
    $text .= '<form action="'. $result["link"].'" method="POST" enctype="multipart/form-
data" class="cart_table_form">
        <input type="hidden" name="module" value="cart">
        <input type="hidden" name="action" value="recalc">
        <input type="hidden" name="ajax" value="">
        <div class="cart_table">;
        $text .= $this->get('table', 'cart', $result);
        $text .= '</div>
        <div class="errors error_table" style="display:none"></div>;
        $text .= '<div class="cart_recalc">;
        // кнопка пересчитать
        $text .= '<span class="button_wrap"><input type="submit" class="button" value="" .
$this->diafan->_('Пересчитать', false) . ""></span>;
        $text .= '</div>;
        $text .= '</form>;
    }
```

2. Подключить файл *modules/cart/cart.js*:

Пример:

```
echo '<script type="text/javascript" src="'. $BASE_PATH.'modules/cart/cart.js"></script>;'
```

Сделать это нужно в файле *modules/cart/views/cart.view.show_block.php*, чтобы JS-файл не подгружался каждый раз при добавлении товара в корзину. При этом не забудьте, что файл *cart.js* подключается также на странице корзины. Чтобы избежать двойного подключения этого файла, нужно либо не подключать его в файле *modules/cart/views/cart.view.form.php*, либо не выводить блок на странице корзины.

Если вам нужно скрыть таблицу товаров в блоке на странице корзины, добавьте следующий код:

Пример:

```
if(! empty($result["rows"]) && $this->diafan->module != 'cart')
{
    $text .= '<form action="'. $result["link"].'" method="POST" enctype="multipart/form-data"
class="cart_table_form">
        <input type="hidden" name="module" value="cart">
        <input type="hidden" name="action" value="recalc">
        <input type="hidden" name="ajax" value="">
        <div class="cart_table">;
        $text .= $this->get('table', 'cart', $result);
    }
```

```

$text .= '</div>
<div class="errors error_table" style="display:none"></div>';
$text .= '<div class="cart_recalc">';
// кнопка пересчитать
$text .= '<span class="button_wrap"><input type="submit" class="button" value="" . $this-
>diafan->_('Пересчитать', false) . ""></span>';
$text .= '</div>';
$text .= '</form>';
}

```

Пример написан для diafan.CMS 5.2. Если возникли проблемы с обновлением таблицы корзины при покупке товаров или пересчете, обновите diafan.CMS с помощью автообновления.

Как добавить виджет комментариев из социальной сети

- [Facebook](#)
- [Вконтакте](#)
- [Оформление.](#)

В примере, рассмотренном ниже, будет показано, как вставить на свой сайт виджеты комментариев популярных социальных сетей «Вконтакте» и «Facebook».

Разместить сам виджет достаточно просто: обе социальные сети предоставляют простой интерфейс, генерирующий код для сайта. Попробуем добавить его на свой сайт.

Facebook

1. Регистрация сайта.

Чтобы поставить виджет от Facebook, для начала нужно зайти на страничку <http://developers.facebook.com/docs/reference/plugins/comments> и заполнить форму, в которой нужно указать адрес вашего сайта, количество комментариев, которые будут отображаться на странице, затем указать ширину виджета и вариант цветовой схемы:



The image shows a web form for configuring a Facebook Comments widget. It has four input fields: 'URL to comment on (?)' with the value 'http://example.com', 'Width (?)' with the value '470', 'Number of posts (?)' with the value '10', and 'Color Scheme (?)' with a dropdown menu set to 'light'. At the bottom is a 'Get Code' button.

Остается нажать на кнопку «Get Code» (получить код).

2. Размещение на сайте.

Полученный код можно разделить на 2 части. Первая, в идеале, ставится после тега `<body>`.

Пример:

```

<div id="fb-root"></div>
<script>(function(d, s, id) {
var js, fjs = d.getElementsByTagName(s)[0];
if (d.getElementById(id)) {return;}
js = d.createElement(s); js.id = id;
js.src = "//connect.facebook.net/ru_RU/all.js#xfbml=1";
fjs.parentNode.insertBefore(js, fjs);
})(document, 'script', 'facebook-jssdk');</script>

```

Вторая часть размещается в том месте, где непосредственно будет вывод блока с комментариями.

Пример:

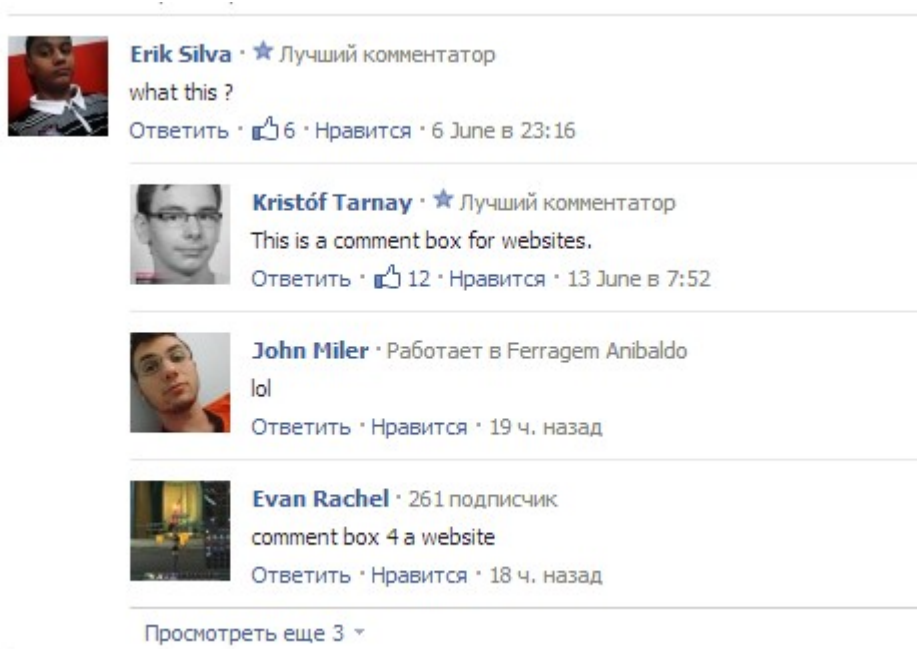
```

<div class="fb-comments" data-href="адрес_сайта" data-num-posts="кол-во_комментов"
data-width="ширина"></div>

```

Добавляем код в шаблон сайта *themes/site.php*.

В результате получается что-то похожее:



Вконтакте

1. Регистрация сайта.

Чтобы разместить сам виджет, для начала необходимо перейти на страничку <http://vk.com/developers.php?oid=-1&p=Comments> и заполнить форму, в которой последовательно указываем сайт (возможно, его нужно будет добавить, если его нет в выпадающем списке), количество отображаемых комментариев, настраиваем тип контента,

доступного для показа, и ширину виджета.

Подключение комментариев к Вашему сайту

Для добавления виджета на Ваш сайт просто скопируйте код для вставки на страницу, куда Вы хотите подключить виджет.

The screenshot shows a web form for configuring the VK Comments widget. It includes fields for site selection, name, address, and domain, followed by a 'Сохранить' (Save) button. Below this, there are radio buttons for the number of comments (5, 10, 15, 20), a checkbox for 'Включить расширенные комментарии' (Enable extended comments), and a width input field set to 496 px. The 'Код для вставки' (Code for insertion) section displays a pre-generated code snippet. At the bottom, there is a link to the full documentation.

Сайт/приложение:

Название сайта:

Адрес сайта:

Основной домен сайта:

Количество комментариев: ☐ 5 ☒ 10 ☐ 15 ☐ 20

Медиа: ☒ Включить расширенные комментарии ([подробнее](#))

Ширина: px

Код для вставки:

```
<script type="text/javascript"
src="//vk.com/js/api/openapi.js?96"></script>

<script type="text/javascript">
  VK.init({apiId: 3715805, onlyWidgets: true});
</script>

<!-- Put this div tag to the place, where the Comments
block will be -->
<div id="vk_comments"></div>
<script type="text/javascript">
  VK.Widgets.Comments("vk_comments", {limit: 10,
width: "496", attach: "*"});
```

[Для более гибкой настройки Вы можете прочитать полную документацию по работе виджета.](#)

Код генерируется автоматически, изменяясь по ходу заполнения формы.

2. Размещение на сайте.

Полученный код делится на 2 части. Первая размещается внутри тегов `<head></head>`.

Пример:

```
<script type="text/javascript">
VK.init({apiId: твой ап, onlyWidgets: true});
</script>
```

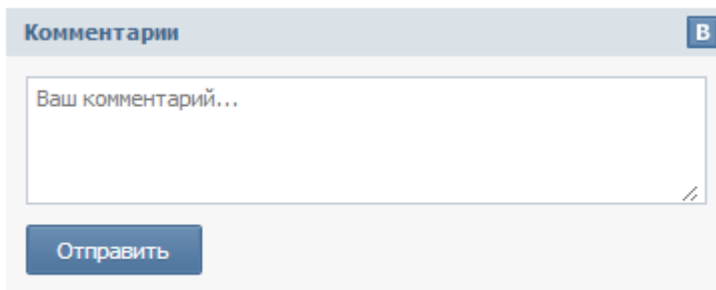
Вторая часть ставится в то место, где будет выводиться контент виджета.

Пример:

```
<div id="vk_comments"></div>
<script type="text/javascript">
VK.Widgets.Comments("vk_comments", {limit: 10, width: "496", attach: ""});
</script>
```

Добавляем код в шаблон сайта *themes/site.php*.

В результате получаем знакомый интерфейс для размещения комментариев:

A screenshot of a web form titled "Комментарии" (Comments). It features a text input field with the placeholder text "Ваш комментарий..." (Your comment...). Below the input field is a blue button labeled "Отправить" (Send). In the top right corner of the form, there is a small blue button with the letter "В".

Оформление.

Оба виджета можно оформить в виде шаблонных тегов. Чтобы сделать это, откроем файл *includes/function.php* и добавим новые функции:

Пример:

```
public function show_facebook()
{
    echo "<div id='fb-root'></div>
    <script>(function(d, s, id) {
    var js, fjs = d.getElementsByTagName(s)[0];
    if (d.getElementById(id)) {return;}
    js = d.createElement(s); js.id = id;
    js.src = '//connect.facebook.net/ru_RU/all.js#xfbml=1';
    fjs.parentNode.insertBefore(js, fjs);
    })(document, 'script', 'facebook-jssdk');</script>
    <div class='fb-comments' data-href='адрес_сайта' data-num-posts='кол-
    во_комментов' data-width='ширина'></div>";
}

public function show_vk()
{
    echo '<script type="text/javascript" src="//vk.com/js/api/openapi.js?63"></script>
    <script type="text/javascript">
        VK.init({apild: тут будет твой ап, onlyWidgets: true});
    </script>';
}
```

```
<div id="vk_comments"></div>
  <script type="text/javascript">
    VK.Widgets.Comments("vk_comments", {limit: 10, width: "496", attach: ""});
  </script>;
}
```

Теперь мы можем использовать шаблонные теги.

Пример:

```
<insert name="show_facebook">
<insert name="show_vk">
```

Справочник шаблонных тегов

Здесь собраны все шаблонные теги, используемые в шаблонах дизайна в diafan.CMS. Теги собраны по алфавиту с группировкой по модулям.

Внимание!

Рекомендуем разработчикам добавить эту страницу в «Избранное».

Теги общего характера

- [show_body](#) – выводит основной контент страницы: заголовок (если не запрещен его вывод в настройке странице «Не показывать заголовок»), текста страницы и прикрепленного модуля. Заменяет три тега: show_h1, show_text, show_module;
- [show_description](#) – выводит мета-тег description страницы;
- [show_h1](#) – выводит заголовок страницы, если не запрещен его вывод в настройке странице «Не показывать заголовок»;
- [show_href](#) – выводит ссылку на страницу сайта. Если текущая страница соответствует адресу, на которую ведет ссылка, то ссылка становится неактивной. Шаблонная функция аналогична стандартной HTML-конструкции `<a href=""></a>`, но ликвидирует ссылки, которые никуда не ведут;
- [show_include](#) – подключает файл-блок шаблона;
- [show_js](#) – подключает JS файлы, обеспечивающие работу diafan.CMS;
- [show_keywords](#) – выводит ключевые слова страницы. Используется для мета-тега keywords;
- [show_module](#) – выводит контент модуля;
- [show_path](#) – выводит навигацию по сайту «Хлебные крошки»;
- [show_protect](#) – выводит JS код запрещающий копирование контента на сайте;
- [show_social_links](#) – выводит ссылки на социальные сети;
- [show_text](#) – выводит текст страницы;
- [show_title](#) – выводит заголовок. Используется между тегами `<title></title>` в шапке сайта;
- [show_year](#) – выводит период функционирования сайта в годах.

On-line консультант

- [show_block](#) – подключает on-line консультант.

Баннеры

- [show_block](#) – выводит баннеры.

Вопрос-ответ

- [show_block](#) – выводит последние вопросы. На странице вопросов, когда выводится

список аналогичных вопросов, блок не выводится;

- [show_block_rel](#) – на странице вопроса выводит похожие вопросы. По умолчанию связи между вопросами являются односторонними, это можно изменить, отметив опцию «В блоке похожих вопросов связь двусторонняя» в настройках модуля;
- [show_form](#) – выводит форму добавления вопроса. Для правильной работы тега должна существовать страница, к которой прикреплен модуль Вопрос-Ответ.

Коды РК

- [show_block](#) – выводит код реферальной ссылки.

Комментарии

- [show_block](#) – выводит последние комментарии.

Магазин

- [show_add_coupon](#) – выводит форму активирования купона на скидку, если неактивированный купон есть в системе, пользователь авторизован и у него не активирован другой купон;
- [show_block](#) – выводит информацию о заказанных товарах, т. н. корзину;
- [show_block](#) – выводит несколько товаров из каталога;
- [show_block_order_rel](#) – товары, которые обычно покупают с текущим товаром;
- [show_block_rel](#) – на странице товара выводит похожие товары. По умолчанию связи между товарами являются односторонними, это можно изменить, отметив опцию «В блоке похожих товаров связь двусторонняя» в настройках модуля;
- [show_search](#) – выводит форму поиска товаров. Если для категорий прикреплены дополнительные характеристики, то поиск по ним производится только на странице категории. Поиск по обязательным полям подключается в настройках модуля (опции «Искать по цене», «Искать по описанию», «Искать по названию», «Искать по артикулу», «Искать товары по акции», «Искать новинки», «Искать хиты»). Если в форму поиска выведены характеристики с типом «выпадающий список» и «список с выбором нескольких значений», то значения характеристик, которые не найдут ни один товар, в форме поиска не выведутся.

Меню

- [show_block](#) – выводит меню.

Новости

- [show_block](#) – выводит последние новости на всех страницах, кроме страницы новостей, когда выводится список тех же новостей, что и в функции;
- [show_block_rel](#) – на странице новости выводит похожие новости. По умолчанию связи между новостями являются односторонними, это можно изменить, отметив опцию «В блоке похожих новостей связь двусторонняя» в настройках модуля;

- [show_calendar](#) – выводит календарь со ссылками на новости за период. Периоды отображаются в виде ссылок на месяцы, только если имеются новости, соответствующие этим периодам.

Обратная связь

- [show_form](#) – выводит форму добавления сообщения. Для правильной работы тега должна существовать страница, к которой прикреплен модуль Обратная связь.

Объявления

- [show_block](#) – выводит последние объявления на всех страницах, кроме страницы объявлений, когда выводится список тех же объявлений, что и в функции;
- [show_block_rel](#) – на странице объявления выводит похожие объявления. По умолчанию связи между объявлениями являются односторонними, это можно изменить, отметив опцию «В блоке похожих объявлений связь двусторонняя» в настройках модуля;
- [show_form](#) – выводит форму добавления сообщения. Для правильной работы тега должна существовать страница, к которой прикреплен модуль Объявления;
- [show_search](#) – выводит форму поиска объявлений. Если для категорий прикреплены дополнительные характеристики, то поиск по ним производится только на странице категории.

Опросы

- [show_block](#) – выводит опросы.

Ошибка на сайте

- [show_block](#) – подключает модуль «Ошибка на сайте».

Поиск по сайту

- [show_search](#) – выводит форму поиска по сайту.

Пользователи

- [show_block](#) – выводит статистику пользователей на сайте;
- [show_login](#) – выводит форму авторизации или приветствие и ссылки на редактирование данных и выход, если пользователь авторизован.

Рассылки

- [show_form](#) – выводит форму подписки на рассылки.

Статьи

- [show_block](#) – выводит последние статьи на всех страницах, кроме страницы статей, когда выводится список тех же статей, что и в функции;
- [show_block_rel](#) – на странице статьи выводит похожие статьи. По умолчанию связи между статьями являются односторонними, это можно изменить, отметив опцию «В блоке похожих статей связь двусторонняя» в настройках модуля.

Страницы сайта

- [show_block](#) – выводит содержимое страницы сайта, номер которой передан в виде атрибута id. Страница должна быть помечена опцией «Блок на сайте» и быть активной;
- [show_comments](#) – выводит комментарии, прикрепленные к странице сайта, если в конфигурации модуля «Страницы сайты» подключены комментарии;
- [show_images](#) – выводит изображения, прикрепленные к странице сайта, если в конфигурации модуля «Страницы сайта» включены параметры «Использовать изображения»;
- [show_links](#) – выводит ссылки на страницы нижнего уровня, принадлежащие текущей странице;
- [show_previous_next](#) – выводит ссылки на предыдущую и последующую страницы;
- [show_tags](#) – выводит теги (слова-якори), прикрепленные к странице сайта, если в конфигурации модуля «Страницы сайты» подключены теги.

Теги

- [show_block](#) – выводит список тегов.

Файловый архив

- [show_block](#) – выводит несколько файлов из файлового архива;
- [show_block_rel](#) – на странице файлы выводит похожие файлы. По умолчанию связи между файлами являются односторонними, это можно изменить, отметив опцию «В блоке похожих файлов связь двусторонняя» в настройках модуля.

Форум

- [show_block](#) – выводит последние темы;
- [show_block_messages](#) – выводит блок сообщений;
- [show_block_rel](#) – на странице темы форума выводит блок похожих тем. Похожие темы подбираются по совпадению слов в заголовках.

Фотогалерея

- [show_block](#) – выводит несколько фотографий;
- [show_block_rel](#) – на странице фотографии выводит похожие фотографии. По

умолчанию связи между фотографиями являются односторонними, это можно изменить, отметив опцию «В блоке похожих фотографий связь двусторонняя» в настройках модуля.

Языки сайта

- [show_block](#) — выводит ссылки на альтернативные языковые версии сайта.

А

show_add_coupon module="shop"

Выводит форму активирования купона на скидку, если неактивированный купон есть в системе, пользователь авторизован и у него не активирован другой купон.

Атрибуты:

— **template** — шаблон тега (файл *modules/shop/views/shop.view.show_add_coupon_***template***.php*; по умолчанию шаблон *modules/shop/views/shop.view.show_add_coupon.php*).

Пример:

```
<insert name="show_add_coupon" module="shop">
```

выведет форму активирования купона

В

show_block module="ab"

Выводит последние объявления на всех страницах, кроме страницы объявлений, когда выводится список тех же объявлений, что и в функции.

Атрибуты:

— **count** — количество выводимых объявлений (по умолчанию 3);

— **site_id** — страницы, к которым прикреплен модуль. Идентификаторы страниц перечисляются через запятую. По умолчанию выбираются все страницы;

— **cat_id** – категории объявлений, если в настройках модуля отмечено «Использовать категории». Идентификаторы категорий перечисляются через запятую. По умолчанию категория не учитывается, выводятся все объявления;

— **sort** – сортировка объявлений: **date** – по дате (по умолчанию), **rand** – в случайном порядке;

— **only_ab** – выводить блок только на странице, к которой прикреплен модуль «Объявления»;

— **images** – количество изображений, прикрепленных к объявления;

— **images_variation** – тег размера изображений, задается в настройках модуля;

— **param** – значения дополнительных характеристик;

Пример:

Объявления обладают следующими характеристиками:

- тип: квартира, дом, участок – выпадающий список, номер 3;
- площадь – число, номер 10;
- новостройка – галочка, номер 16.

Значит значение атрибута *param3=5¶m3=6¶m10>50¶m16=0* расшифровывается как объявления о квартире и доме (5 и 6 номер), площадью более 50 кв.м, в новостройке.

— **template** – шаблон тега (файл *modules/ab/views/ab.view.show_block_template.php*; по умолчанию шаблон *modules/ab/views/ab.view.show_block.php*).

Пример:

```
<insert name="show_block" module="ab">
```

выведет 3 последние объявления сайта

```
<insert name="show_block" module="ab" count="5">
```

выведет 5 последних объявлений сайта

```
<insert name="show_block" module="ab" count="1" site_id="12">
```

выведет последнее объявление, прикрепленное к странице 12

show_block module="bs"

Выводит баннеры.

Атрибуты:

- **count** – количество выводимых баннеров. По умолчанию 1. Значение **all** выведет все баннеры;
- **id** – идентификатор баннера, если задан, атрибут **count** игнорируется;
- **sort** – сортировка баннеров: **date** – по дате (по умолчанию), **rand** – в случайном порядке;
- **cat_id** – категория баннеров, если в настройках модуля отмечено «Использовать категории»;
- **template** – шаблон тега (файл *modules/bs/views/bs.view.show_block_template.php*; по умолчанию шаблон *modules/bs/views/bs.view.show_block.php*).

Пример:

```
<insert name="show_block" module="bs">
```

выведет 1 случайный баннер

show_block module="cart"

Выводит информацию о заказанных товарах, т. н. корзину.

Атрибуты:

- **template** – шаблон тега (файл *modules/cart/views/cart.view.show_block_template.php*; по умолчанию шаблон *modules/cart/views/cart.view.show_block.php*).

Пример:

```
<insert name="show_block" module="cart">
```

выведет информацию о корзине

show_block module="clauses"

Выводит последние статьи на всех страницах, кроме страницы статей, когда выводится список тех же статей, что и в функции.

Атрибуты:

- **count** – количество выводимых статей (по умолчанию 3);

- **site_id** – страницы, к которым прикреплен модуль. Идентификаторы страниц перечисляются через запятую. По умолчанию выбираются все страницы;
- **cat_id** – категории статей, если в настройках модуля отмечено «Использовать категории». Идентификаторы категорий перечисляются через запятую. По умолчанию категория не учитывается, выводятся все статьи;
- **images** – количество изображений, прикрепленных к статье;
- **images_variation** – тег размера изображений, задается в настройках модуля;
- **sort** – сортировка статей: по умолчанию как на странице модуля, **date** – по дате, **rand** – в случайном порядке, **keywords** – статьи, похожие по названию для текущей страницы;
- **template** – шаблон тега (файл *modules/clauses/views/clauses.view.show_block_template.php*; по умолчанию шаблон *modules/clauses/views/clauses.view.show_block.php*).

Пример:

```
<insert name="show_block" module="clauses">
```

выведет 3 статьи сайта

```
<insert name="show_block" module="clauses" count="5" sort="date">
```

выведет 5 последних статей сайта

```
<insert name="show_block" module="clauses" count="1" cat_id="35">
```

выведет одну статью сайта, из раздела статей с id=35

show_block module="comments"

Выводит последние комментарии.

Атрибуты:

- **count** – количество выводимых комментариев (по умолчанию 3);
- **modules** – модули (разделенные запятой), к которым подключены комментарии;
- **element_id** – идентификаторы элементов модулей (разделенные запятой), к которым подключены комментарии;
- **sort** – сортировка комментариев: **date** – по дате (по умолчанию), **rand** – в случайном

порядке;

— **template** – шаблон тега (файл *modules/comments/views/comments.view.show_block_template.php*; по умолчанию шаблон *modules/comments/views/comments.view.show_block.php*).

Пример:

```
<insert name="show_block" module="comments" modules="shop" count="2">
```

выведет 2 последних комментария к товарам

show_block module="consultant"

Подключает on-line консультант.

Атрибуты:

— **template** – шаблон тега (файл *modules/consultant/views/consultant.view.show_block_template.php*; по умолчанию шаблон *modules/consultant/views/consultant.view.show_block.php*).

Пример:

```
<insert name="show_block" module="consultant">
```

выведет код для подключения on-line консультанта

show_block module="faq"

Выводит последние вопросы. На странице вопросов, когда выводится список аналогичных вопросов, блок не выводится.

Атрибуты:

— **count** – количество выводимых вопросов (по умолчанию 3);

— **site_id** – страницы, к которым прикреплен модуль. Идентификаторы страниц перечисляются через запятую. По умолчанию выбираются все страницы;

— **cat_id** – категории вопросов, если в настройках модуля отмечено «Использовать категории». Идентификаторы категорий перечисляются через запятую. По умолчанию категория не учитывается, выводятся все вопросы;

— **often** – часто задаваемые вопросы (если **often**=1, выводятся вопросы с пометкой «Часто задаваемый вопрос»);

— **sort** – сортировка вопросов: **date** – по дате (по умолчанию), **rand** – в случайном порядке;

— **template** – шаблон тега (файл *modules/faq/views/faq.view.show_block_template.php*; по умолчанию шаблон *modules/faq/views/faq.view.show_block.php*).

Формат даты задается в настройках модуля.

Пример:

```
<insert name="show_block" module="faq">
```

выведет 3 последних активных вопроса сайта с ответами

```
<insert name="show_block" module="faq" count="5">
```

выведет в ячейке таблицы 5 последних активных вопросов сайта с ответами

```
<insert name="show_block" module="faq" often="1">
```

выведет часто задаваемые вопросы

show_block module="files"

Выводит несколько файлов из файлового архива.

Атрибуты:

- **count** – количество выводимых файлов (по умолчанию 3);
- **site_id** – страницы, к которым прикреплен модуль. Идентификаторы страниц перечисляются через запятую. По умолчанию выбираются все страницы;
- **cat_id** – категории файлов, если в настройках модуля отмечено «Использовать категории». Идентификаторы категорий перечисляются через запятую. По умолчанию категория не учитывается, выводятся все файлы;
- **images** – количество изображений, прикрепленных к файлу;
- **images_variation** – тег размера изображений, задается в настройках модуля;
- **sort** – сортировка файлов: по умолчанию как на странице модуля, **rand** – в случайном порядке, **date** – по дате;
- **template** – шаблон тега (файл *modules/files/views/files.view.show_block_template.php*; по умолчанию шаблон *modules/files/views/files.view.show_block.php*).

Пример:

```
<insert name="show_block" module="files">
```

выведет 3 последние файла из файлового архива

show_block module="forum"

Выводит последние темы.

Атрибуты:

- **count** – количество выводимых тем (по умолчанию 3);
- **cat_id** – категория или раздел форума (id категории, по умолчанию выводятся все темы без учета категории);
- **template** – шаблон тега (файл *modules/forum/views/forum.view.show_block_***template***.php*; по умолчанию шаблон *modules/forum/views/forum.view.show_block.php*).

Пример:

```
<insert name="show_block_rel" module="forum">
```

выведет 3 похожие на текущую темы

show_block module="languages"

Выводит ссылки на альтернативные языковые версии сайта.

Атрибуты:

- **template** – шаблон тега (файл *modules/languages/views/languages.view.show_block_***template***.php*; по умолчанию шаблон *modules/languages/views/languages.view.show_block.php*).

Пример:

```
<insert name="show_block" module="languages">
```

выведет список ссылок на языковые версии сайта, «ru/eng»

show_block module="menu"

Выводит меню.

Атрибуты:

- **id** – идентификатор категории меню;

— **template** – имя шаблона;

Внимание!

Атрибут **template** является «переключателем» режима работы шаблонного тега. Если в шаблонном теге есть этот атрибут, то меню на сайте оформляется с помощью традиционных шаблонов модулей. Если этого атрибута нет, то меню на сайте оформляется с помощью атрибутов оформления.

Если в атрибуте **template** указано какое-либо значение, то должен существовать файл *modules/menu/view/menu.view.show_block_***template***.php*.

Пример:

```
<insert name="show_block" module="menu" template="supermenu">
```

меню оформится с помощью файла *modules/menu/view/menu.view.show_block_***supermenu***.php*, который необходимо будет создать (можно скопировать любой другой существующий файл шаблона)

Значение **template="default"** подключит шаблон *modules/menu/view/menu.view.show_block.php*.

Значение **template="select"** подключит шаблон выбранный для меню в административном интерфейсе – параметр «Шаблон вывода меню».

Если атрибут **template** не задан, меню оформляется с помощью атрибутов оформления и выводится шаблоном *modules/menu/view/menu.view.show_menu.php*.

— **tag_start_номер** – текст, выводимый перед каждой ссылкой меню (может содержать слово **Increment** – при выводе автоматически заменяемое на арифметическую прогрессию, 1, 2, 3, ... и **Level** – заменяемое на номер уровня меню);

— **tag_end_номер** – текст, выводимый после каждой ссылки пункта меню (может содержать слово **Increment** - заменяемое на увеличивающийся номер и **Level** – заменяемое на номер уровня меню);

— **tag_level_start_номер** – текст, выводимый перед уровнем (номер) меню;

— **tag_level_end_номер** – текст, выводимый после уровня (номер) меню;

— **tag_active_start_номер** – текст, выводимый перед активным пунктом меню уровня (может содержать слово **Increment** - заменяемое на увеличивающийся номер и **Level** – заменяемое на номер уровня меню);

— **tag_active_end_номер** – текст, выводимый после активного пункта меню уровня (может содержать слово **Increment** - заменяемое на увеличивающийся номер и **Level** – заменяемое на номер уровня меню);

— **tag_active_child_start_номер** – текст, выводимый перед пунктом меню уровня (номер) с активным дочерним пунктом (может содержать слово **Increment** - заменяемое на увеличивающийся номер и **Level** – заменяемое на номер уровня меню);

— **tag_active_child_end_номер** – текст, выводимый после пункта меню уровня (номер) с активным дочерним пунктом (может содержать слово **Increment** - заменяемое на увеличивающийся номер и **Level** – заменяемое на номер уровня меню);

— **separator_номер** – текст, разделяющий пункты меню.

Номер в названии атрибутов обозначает номер уровня. Если атрибуты не указаны для текущего уровня, то они наследуются у верхнего уровня.

Пример:

```
<insert name="show_block" module="menu">
```

выведет просто список ссылок основного меню

```
<table>
  <insert name="show_block" module="menu" id="2" tag_start_1="[tr][td]" tag_end_1="[/td]
[/tr]">
</table>
```

сформирует таблицу, в отдельной ячейке которой будут пункты меню из второго меню

```
<insert name="show_block" module="menu" tag_start_1="[div class=`menu`]"
tag_end_1="[/div]" tag_start_2="[div class=`submenu`]" tag_end_2="[/div]">
```

Выведет:

```
<div class="menu"><a href="...">О нас</a></div>
  <div class="submenu"><a href="...">Животные</a></div>
  <div class="submenu"><a href="...">Миссия</a></div>
  <div class="submenu"><a href="...">История</a></div>
<div class="menu"><a href="...">Фотогалерея</a></div>
```

```
<insert name="show_block" module="menu" tag_start_1="[div class=`menuLevel`]"
tag_end_1="[/div]">
```

Выведет:

```
<div class="menu1"><a href="...">О нас</a></div>
  <div class="menu2"><a href="...">Животные</a></div>
  <div class="menu2"><a href="...">Миссия</a></div>
  <div class="menu2"><a href="...">История</a></div>
<div class="menu1"><a href="...">Фотогалерея</a></div>
```

show_block module="mistakes"

Подключает модуль «Ошибка на сайте».

Пример:

```
<insert name="show_block" module="mistakes">
```

Выведет «Если Вы заметили ошибку на сайте, выделите ее и нажмите Ctrl+Enter».

Шаблон: файл *modules/mistakes/views/mistakes.show_block.php*.

show_block module="news"

Выводит последние новости на всех страницах, кроме страницы новостей, когда выводится список тех же новостей, что и в функции.

Атрибуты:

- **count** – количество выводимых новостей (по умолчанию 3);
- **site_id** – страницы, к которым прикреплен модуль. Идентификаторы страниц перечисляются через запятую. По умолчанию выбираются все страницы;
- **cat_id** – категории новостей, если в настройках модуля отмечено «Использовать категории»;
- **sort** – сортировка новостей: **date** – по дате (по умолчанию), **keywords** – новости, похожие по названию для текущей страницы;
- **images** – количество изображений, прикрепленных к новости;
- **images_variation** – тег размера изображений, задается в настройках модуля;
- **template** – шаблон тега (файл *modules/news/views/news.view.show_block_template.php*; по умолчанию шаблон *modules/news/views/news.view.show_block.php*).

Формат даты задается в настройках модуля.

Пример:

```
<insert name="show_block" module="news">
```

выведет 3 последние новости сайта

```
<insert name="show_block" module="news" count="5">
```

выведет 5 последних новостей сайта

```
<insert name="show_block" module="news" count="1" site_id="12">
```

выведет последнюю новость, прикрепленную к странице 12

show_block module="photo"

Выводит несколько фотографий.

Атрибуты:

- **count** – количество выводимых фотографий (по умолчанию 3);
- **site_id** – страницы, к которым прикреплен модуль. Идентификаторы страниц перечисляются через запятую. По умолчанию выбираются все страницы;
- **cat_id** – альбомы фотографий, если в настройках модуля отмечено «Использовать альбомы». Идентификаторы альбомов перечисляются через запятую. По умолчанию альбом не учитывается, выводятся все фотографии;
- **sort** – сортировка фотографий: по умолчанию как на странице модуля, **date** – по дате, **rand** – в случайном порядке;
- **images_variation** – тег размера изображений, задается в настройках модуля;
- **template** – шаблон тега (файл *modules/photo/views/photo.view.show_block_template.php*; по умолчанию шаблон *modules/photo/views/photo.view.show_block.php*).

Пример:

```
<insert name="show_block" module="photo">
```

выведет 3 последние фотографии из фотогалереи

show_block module="referrals"

Выводит код реферальной ссылки.

Атрибуты:

- **show_referral_id** – выводить код реферальной ссылки ("true"|"false");
- **template** – шаблон тега (файл *modules/referrals/views/referrals.view.show_block_template.php*; по умолчанию шаблон *modules/referrals/views/referrals.view.show_block.php*).

Пример:

```
<insert name="show_block" module="referrals">
```

show_block module="shop"

Выводит несколько товаров из каталога.

Атрибуты:

- **count** – количество выводимых товаров (по умолчанию 3);
- **site_id** – страницы, к которым прикреплен модуль. Идентификаторы страниц перечисляются через запятую. По умолчанию выбираются все страницы;
- **cat_id** – категории товаров, если в настройках модуля отмечено «Использовать категории». Идентификаторы категорий перечисляются через запятую. По умолчанию категория не учитывается, выводятся все товары;
- **images** – количество изображений, прикрепленных к товару;
- **images_variation** – тег размера изображений, задается в настройках модуля;
- **sort** – сортировка товаров: по умолчанию как на странице модуля, **date** – по дате, **rand** – в случайном порядке, **price** – по цене, **sale** – по количеству продаж;
- **only_shop** – выводить блок только на странице, к которой прикреплен модуль «Магазин»;
- **hits_only** – выводить только товары с пометкой «Хит»;
- **action_only** – выводить только товары с пометкой «Акция»;
- **new_only** – выводить только товары с пометкой «Новинка»;
- **param** – значения дополнительных характеристики;

Пример:

Товары обладают следующими характеристиками:

- цвет – выпадающий список, номер 3;
- высота – число, номер 10;
- наличие аналогов – галочка, номер 16.

Значит значение атрибута *param3=5&3=6&10>12&16=0* расшифровывается как товары красного и синего цвета (5 и 6 номер), высотой более 12, не имеющие

аналогов.

— **template** – шаблон тега (файл *modules/shop/views/shop.view.show_block_template.php*; по умолчанию шаблон *modules/shop/views/shop.view.show_block.php*);

— **discount_only** – выводить только товары, на которые действует скидка.

Пример:

```
<insert name="show_block" module="shop">
```

выведет 3 последних товара из магазина

```
<insert name="show_block" module="shop" count="5" sort="rand">
```

выведет 5 случайных товаров из магазина

```
<insert name="show_block" module="shop" sort="price" count="4" cat_id="12">
```

выведет 4 самых дешевых товаров из рубрики №12 магазина

show_block module="site"

Выводит содержимое страницы сайта, номер которой передан в виде атрибута id. Страница должна быть помечена опцией «Блок на сайте» и быть активной.

Атрибуты:

— **id** – идентификатор блока;

— **template** – шаблон тега (файл *modules/site/views/site.view.show_block_template.php*; по умолчанию шаблон *modules/site/views/site.view.show_block.php*).

Пример:

```
<insert name="show_block" module="site" id="7">
```

выведет в ячейке таблицы содержимое страницы с id=7, редактируемой в модуле «Страницы сайта»

show_block module="tags"

Выводит список тегов.

Атрибуты:

— **template** – шаблон тега (файл *modules/tags/views/tags.view.show_block_template.php*; по умолчанию шаблон *modules/tags/views/tags.view.show_block.php*).

Пример:

```
<insert name="show_block" module="tags">
```

выведет облако тегов

show_block module="users"

Выводит статистику пользователей на сайте.

Атрибуты:

— **template** – шаблон тега (файл *modules/users/views/users.view.show_block_template.php*; по умолчанию шаблон *modules/users/views/users.view.show_block.php*).

Пример:

```
<insert name="show_block" module="users">
```

выведет «Сейчас на сайте: 13 гостей, 9 пользователей.»

show_block module="votes"

Выводит опросы.

Атрибуты:

— **id** – задает номер вопроса;

— **count** – количество вопросов. Значение all выведет все вопросы. По умолчанию 1;

— **template** – шаблон тега (файл *modules/votes/views/votes.view.show_block_template.php*; по умолчанию шаблон *modules/votes/views/votes.view.show_block.php*).

Пример:

```
<insert name="show_block" module="votes">
```

выведет вопрос, варианты ответа и кнопки «Ответить» и «Результаты», согласно настройкам и заполнению модуля «Опросы» в административной части сайта

show_block_messages module="forum"

Выводит блок сообщений.

Атрибуты:

- **count** – количество выводимых сообщений;
- **cat_id** – раздел форума;
- **template** – шаблон тега (файл *modules/forum/views/forum.view.show_block_messages_***template***.php*; по умолчанию шаблон *modules/forum/views/forum.view.show_block_messages.php*).

Пример:

```
<insert name="show_block_messages" module="forum">
```

show_block_order_rel module="shop"

Товары, которые обычно покупают с текущим товаром.

Атрибуты:

- **count** – количество выводимых товаров (по умолчанию 3);
- **images** – количество изображений, прикрепленных к товару;
- **images_variation** – тег размера изображений, задается в настройках модуля;
- **template** – шаблон тега (файл *modules/shop/views/shop.view.show_block_order_rel_***template***.php*; по умолчанию шаблон *modules/shop/views/shop.view.show_block_order_rel.php*).

Пример:

```
<insert name="show_block_order_rel" module="shop">
```

выведет 3 товара, которые обычно покупают с текущим товаром

show_block_rel module="ab"

На странице объявления выводит похожие объявления. По умолчанию связи между объявлениями являются односторонними, это можно изменить, отметив опцию «В блоке похожих объявлений связь двусторонняя» в настройках модуля.

Атрибуты:

- **count** – количество выводимых объявлений (по умолчанию 3);

- **images** – количество изображений, прикрепленных к объявления;
- **images_variation** – тег размера изображений, задается в настройках модуля;
- **template** – шаблон тега (файл *modules/ab/views/ab.view.show_block_rel_***template***.php*; по умолчанию шаблон *modules/ab/views/ab.view.show_block_rel.php*).

Формат даты задается в настройках модуля.

Пример:

```
<insert name="show_block_rel" module="ab">
```

выведет 3 объявления, прикрепленные к текущей объявления

show_block_rel module="clauses"

На странице статьи выводит похожие статьи. По умолчанию связи между статьями являются односторонними, это можно изменить, отметив опцию «В блоке похожих статей связь двусторонняя» в настройках модуля.

Атрибуты:

- **count** – количество выводимых статей (по умолчанию 3);
- **images** – количество изображений, прикрепленных к статье;
- **images_variation** – тег размера изображений, задается в настройках модуля;
- **template** – шаблон тега (файл *modules/clauses/views/clauses.view.show_block_rel_***template***.php*; по умолчанию шаблон *modules/clauses/views/clauses.view.show_block_rel.php*).

Формат даты задается в настройках модуля.

Пример:

```
<insert name="show_block_rel" module="clauses">
```

выведет 3 статьи, прикрепленные к текущей статье

show_block_rel module="faq"

На странице вопроса выводит похожие вопросы. По умолчанию связи между вопросами являются односторонними, это можно изменить, отметив опцию «В блоке похожих вопросов связь двусторонняя» в настройках модуля.

Атрибуты:

— **count** – количество выводимых вопросов (по умолчанию 3);

— **template** – шаблон тега (файл *modules/faq/views/faq.view.show_block_rel_***template***.php*; по умолчанию шаблон *modules/faq/views/faq.view.show_block_rel.php*).

Формат даты задается в настройках модуля.

Пример:

```
<insert name="show_block_rel" module="faq">
```

выведет 3 вопроса, прикрепленные к текущему вопросу

show_block_rel module="files"

На странице файлы выводит похожие файлы. По умолчанию связи между файлами являются односторонними, это можно изменить, отметив опцию «В блоке похожих файлов связь двусторонняя» в настройках модуля.

Атрибуты:

— **count** – количество выводимых файлов (по умолчанию 3);

— **images** – количество изображений, прикрепленных к файлу;

— **images_variation** – тег размера изображений, задается в настройках модуля;

— **template** – шаблон тега (файл *modules/files/views/files.view.show_block_rel_***template***.php*; по умолчанию шаблон *modules/files/views/files.view.show_block_rel.php*).

Пример:

```
<insert name="show_block_rel" module="files">
```

выведет 3 файла, прикрепленные к текущему файлу

show_block_rel module="forum"

На странице темы форума выводит блок похожих тем. Похожие темы подбираются по совпадению слов в заголовках.

Атрибуты:

— **count** – количество выводимых тем;

— **cat_id** – раздел форума;

— **template** – шаблон тега (файл *modules/forum/views/forum.view.show_block_rel_template.php*; по умолчанию шаблон *modules/forum/views/forum.view.show_block_rel.php*).

Пример:

```
<insert name="show_block_rel" module="forum">
```

выведет 3 похожие на текущую темы

show_block_rel module="news"

На странице новости выводит похожие новости. По умолчанию связи между новостями являются односторонними, это можно изменить, отметив опцию «В блоке похожих новостей связь двусторонняя» в настройках модуля.

Атрибуты:

- **count** – количество выводимых новостей (по умолчанию 3);
- **images** – количество изображений, прикрепленных к новости;
- **images_variation** – тег размера изображений, задается в настройках модуля;
- **template** – шаблон тега (файл *modules/news/views/news.view.show_block_rel_template.php*; по умолчанию шаблон *modules/news/views/news.view.show_block_rel.php*).

Формат даты задается в настройках модуля.

Пример:

```
<insert name="show_block_rel" module="news">
```

выведет 3 новости, прикрепленные к текущей новости

show_block_rel module="photo"

На странице фотографии выводит похожие фотографии. По умолчанию связи между фотографиями являются односторонними, это можно изменить, отметив опцию «В блоке похожих фотографий связь двусторонняя» в настройках модуля.

Атрибуты:

- **count** – количество выводимых фотографий (по умолчанию 3);
- **images_variation** – тег размера изображений, задается в настройках модуля;
- **template** – шаблон тега (файл

*modules/photo/views/photo.view.show_block_rel_***template***.php*; по умолчанию шаблон *modules/photo/views/photo.view.show_block_rel.php*).

Пример:

```
<insert name="show_block_rel" module="photo">
```

выведет 3 фотографии, прикрепленные к текущей фотографии

show_block_rel module="shop"

На странице товара выводит похожие товары. По умолчанию связи между товарами являются односторонними, это можно изменить, отметив опцию «В блоке похожих товаров связь двусторонняя» в настройках модуля.

Атрибуты:

- **count** – количество выводимых товаров (по умолчанию 3);
- **images** – количество изображений, прикрепленных к товару;
- **images_variation** – тег размера изображений, задается в настройках модуля;
- **template** – шаблон тега (файл *modules/shop/views/shop.view.show_block_rel_***template***.php*; по умолчанию шаблон *modules/shop/views/shop.view.show_block_rel.php*).

Пример:

```
<insert name="show_block_rel" module="shop">
```

выведет 3 товара, прикрепленные к текущему товару

show_body

Выводит основной контент страницы: заголовка (если не запрещен его вывод в настройке странице «Не показывать заголовок»), текста страницы и прикрепленного модуля. Заменяет три тега: `show_h1`, `show_text`, `show_module`.

Пример:

```
<insert name="show_body">
```

Шаблон: функция `show_body` в файле *includes/function.php*.

C

show_calendar module="news"

Выводит календарь со ссылками на новости за период. Периоды отображаются в виде ссылок на месяцы, только если имеются новости, соответствующие этим периодам.

Атрибуты:

- **site_id** – страница, к которой прикреплен модуль, по умолчанию выбирается одна страница;
- **cat_id** – категория новостей (id категории, по умолчанию учитываются все новости), если в настройках модуля отмечено «Использовать категории»;
- **only_news** – выводить блок только на странице «Новости»;
- **detail** – детализация (**day** – дни, **month** – месяца (по умолчанию), **year** – годы);
- **template** – шаблон тега (файл *modules/news/views/news.view.show_calendar_template.php* или *modules/news/views/news.view.show_calendar_day_template.php* для детализации по дням; по умолчанию шаблон *modules/news/views/news.view.show_calendar.php* или *modules/news/views/news.view.show_calendar_day.php* для детализации по дням).

Пример:

```
<insert name="show_calendar" module="news">
```

выведет календарь новостей

show_comments module="site"

Выводит комментарии, прикрепленные к странице сайта, если в конфигурации модуля «Страницы сайты» подключены комментарии.

Атрибуты:

- **template** – шаблон тега (файл *modules/site/views/site.view.show_comments_template.php*; по умолчанию шаблон *modules/site/views/site.view.show_comments.php*).

Пример:

```
<insert name="show_comments" module="site">
```

выведет форму для отправки комментариев и добавленные комментарии посетителей

D

show_description

Выводит мета-тег description страницы.

Пример:

```
<meta name="description" content="<insert name="show_description">">
```

выведет описание текущей страницы

Шаблон: функция show_description в файле *includes/function.php*.

F

show_form module="ab"

Выводит форму добавления сообщения. Для правильной работы тега должна существовать страница, к которой прикреплен модуль Объявления.

Атрибуты:

- **only_ab** – выводить форму поиска только на странице модуля;
- **site_id** – страницы, к которым прикреплен модуль. Идентификаторы страниц перечисляются через запятую. По умолчанию выбираются все страницы. Если задано несколько страниц, то в форме появляется выпадающий список «Раздел»;
- **cat_id** – категории объявлений, если в настройках модуля отмечено «Использовать категории». Идентификаторы категорий перечисляются через запятую. Можно указать значение **current**, тогда форма будет добавлять объявление в текущую (открытую) категорию или выводить поле «Категория», если ни одна категория не открыта. Если задано несколько категорий, то в форме появляется выпадающий список «Категория»;
- **template** – шаблон тега (файл *modules/ab/views/feedback.view.form_template.php*; по умолчанию шаблон *modules/ab/views/ab.view.form.php*).

Пример:

```
<insert name="show_form" module="ab">
```


выведет форму добавления объявления

show_form module="faq"

Выводит форму добавления вопроса. Для правильной работы тега должна существовать страница, к которой прикреплен модуль Вопрос-Ответ.

Атрибуты:

- **site_id** – страница, к которой прикреплен модуль, по умолчанию выбирается одна страница;
- **cat_id** – категория вопросов (id категории, по умолчанию выбирается одна категория), если в настройках модуля отмечено «Использовать категории»;
- **template** – шаблон тега (файл *modules/faq/views/faq.view.form_***template***.php*; по умолчанию шаблон *modules/faq/views/faq.view.form.php*).

Пример:

```
<insert name="show_form" module="faq">
```

выведет форму добавления вопроса

show_form module="feedback"

Выводит форму добавления сообщения. Для правильной работы тега должна существовать страница, к которой прикреплен модуль Обратная связь.

Атрибуты:

- **site_id** – страница, к которой прикреплен модуль, по умолчанию выбирается одна страница;
- **template** – шаблон тега (файл *modules/feedback/views/feedback.view.form_***template***.php*; по умолчанию шаблон *modules/feedback/views/feedback.view.form.php*).

Пример:

```
<insert name="show_form" module="feedback">
```

выведет форму добавления сообщения

show_form module="subscription"

Выводит форму подписки на рассылки.

Атрибуты:

— **template** – шаблон тега (файл *modules/subscription/views/subscription.view.form_***template***.php*; по умолчанию шаблон *modules/subscription/views/subscription.view.form.php*).

Пример:

```
<insert name="show_form" module="subscription">
```

выведет форму подписки на рассылки

Н

show_h1

Выводит заголовок страницы, если не запрещен его вывод в настройке странице «Не показывать заголовок».

Пример:

```
<h1><insert name="show_h1"></h1>
```

выведет <h1>Название страницы</h1>

Шаблон: функция show_h1 в файле *includes/function.php*.

show_href

Выводит ссылку на страницу сайта. Если текущая страница соответствует адресу, на которую ведет ссылка, то ссылка становится неактивной. Шаблонная функция аналогична стандартной HTML-конструкции `<a href=""></a>`, но ликвидирует ссылки, которые никуда не ведут.

Атрибуты:

— **rewrite** – псевдоссылка страницы, на которую ведет ссылка, например: news;

— **img** – адрес изображения, используемого в качестве ссылки, например: img/logo.gif;

— **img_act** – адрес изображения на текущей странице, например: img/home_act.gif;

— **width** – ширина изображения;

— **height** – высота изображения;

— **alt** – альтернативный тег для изображения или текст ссылки.

Пример:

```
<insert name="show_href">
```

Шаблон: функция `show_href` в файле `includes/function.php`.

I

show_images module="site"

Выводит изображения, прикрепленные к странице сайта, если в конфигурации модуля «Страницы сайта» включены параметры «Использовать изображения».

Атрибуты:

— **template** – шаблон тега (файл `modules/site/views/site.view.show_images_`**template**`.php`; по умолчанию шаблон `modules/site/views/site.view.show_images.php`).

Пример:

```
<insert name="show_images" module="site">
```

выведет в ячейке таблицы группу фотографий, прикрепленных к текущей странице

show_include

Подключает файл-блок шаблона.

Атрибуты:

— **file** – имя файла из папки `themes/blocks`.

Пример:

```
<insert name="show_include" file="diafan">
```

подключит файл `themes/blocks/diafan.php`

Шаблон: функция `show_include` в файле `includes/function.php`.

Ж

show_js

Подключает JS файлы, обеспечивающие работу diafan.CMS.

Пример:

```
<insert name="show_js">
```

Шаблон: функция show_js в файле *includes/function.php*.

К

show_keywords

Выводит ключевые слова страницы. Используется для мета-тега keywords.

Пример:

```
<meta name="keywords" content="<insert name="show_keywords">">
```

выведет ключевые слова текущей страницы

Шаблон: функция show_keywords в файле *includes/function.php*.

Л

show_links module="site"

Выводит ссылки на страницы нижнего уровня, принадлежащие текущей странице.

Атрибуты:

— **template** – шаблон тега (файл *modules/site/views/site.view.show_links_***template***.php*;
по умолчанию шаблон *modules/site/views/site.view.show_links.php*).

Пример:

```
<insert name="show_links" module="site">
```

show_login module="registration"

Выводит форму авторизации или приветствие и ссылки на редактирование данных и выход, если пользователь авторизован.

Атрибуты:

— **template** – шаблон тега (файл *modules/registration/views/registration.view.show_login_***template***.php*; по умолчанию шаблон *modules/registration/views/registration.view.show_login.php*).

Пример:

```
<insert name="show_login" module="registration">
```

выведет форму авторизации

M

show_module

Выводит контент модуля.

Пример:

```
<insert name="show_module">
```

Шаблон: функция `show_module` в файле *includes/function.php*.

P

show_path

Выводит навигацию по сайту «Хлебные крошки».

Атрибуты:

— **separator** – разделитель ссылок в навигации.

Пример:

```
<insert name="show_path" separator="/">
```

выведет ссылки навигации по сайту, «Главная / О компании / Услуги»

Шаблон: функция `show_path` в файле `includes/function.php`.

show_previous_next module="site"

Выводит ссылки на предыдущую и последующую страницы.

Атрибуты:

— **template** – шаблон тега (файл `modules/site/views/site.view.show_previous_next_`**template**`.php`; по умолчанию шаблон `modules/site/views/site.view.show_previous_next.php`).

Пример:

```
<insert name="show_previous_next" module="site">
```

show_protect

Выводит JS код запрещающий копирование контента на сайте.

Пример:

```
<body<insert name="show_protect">>
```

Шаблон: функция `show_protect` в файле `includes/function.php`.

S

show_search module="ab"

Выводит форму поиска объявлений. Если для категорий прикреплены дополнительные характеристики, то поиск по ним производится только на странице категории.

Атрибуты:

— **only_ab** – выводить форму поиска только на странице модуля;

— **site_id** – страницы, к которым прикреплен модуль. Идентификаторы страниц

перечисляются через запятую. По умолчанию выбираются все страницы. Если выбрано несколько страниц сайта, то в форме поиска появляется выпадающий список по выбранным страницам;

— **cat_id** – категории объявлений, если в настройках модуля отмечено «Использовать категории». Идентификаторы категорий перечисляются через запятую. Можно указать значение **current**, тогда поиск будет осуществляться по текущей (открытой) категории или по всем категориям, если ни одна категория не открыта. Если выбрано несколько категорий, то в форме поиска появится выпадающий список категорий, который будет подгружать прикрепленные к категориям характеристики. Можно указать значение **all**, тогда поиск будет осуществляться по всем категориям объявлений и в форме будут участвовать только общие характеристики;

— **ajax** – подгружать результаты поиска без перезагрузки страницы. Результаты подгружаются только если открыта страница со списком объявлений, иначе поиск работает обычным образом;

— **template** – шаблон тега (файл *modules/ab/views/ab.view.show_search_template.php*; по умолчанию шаблон *modules/ab/views/ab.view.show_search.php*).

Пример:

```
<insert name="show_search" module="ab">
```

выведет форму поиска по каталогу объявлений

show_search module="search"

Выводит форму поиска по сайту.

Атрибуты:

— **button** – значение кнопки «Найти». Для неосновной языковой версии значение можно перевести в административной части в меню «Языки сайта» – «Перевод интерфейса»;

— **template** – шаблон тега (файл *modules/search/views/search.view.show_search_template.php*; по умолчанию шаблон *modules/search/views/search.view.show_search.php*).

Пример:

```
<insert name="show_search" module="search" button="найти">
```

выведет форму поиска по сайту. "button" - необязательный атрибут названия кнопки поиска

show_search module="shop"

Выводит форму поиска товаров. Если для категорий прикреплены дополнительные

характеристики, то поиск по ним производится только на странице категории. Поиск по обязательным полям подключается в настройках модуля (опции «Искать по цене», «Искать по описанию», «Искать по названию», «Искать по артикулу», «Искать товары по акции», «Искать новинки», «Искать хиты»). Если в форму поиска выведены характеристики с типом «выпадающий список» и «список с выбором нескольких значений», то значения характеристик, которые не найдут ни один товар, в форме поиска не выведутся.

Атрибуты:

— **only_shop** – выводить форму поиска только на странице магазина;

— **site_id** – страницы, к которым прикреплен модуль. Идентификаторы страниц перечисляются через запятую. По умолчанию выбираются все страницы. Если выбрано несколько страниц сайта, то в форме поиска появляется выпадающий список по выбранным страницам;

— **cat_id** – категории товаров, если в настройках модуля отмечено «Использовать категории». Идентификаторы категорий перечисляются через запятую. Можно указать значение **current**, тогда поиск будет осуществляться по текущей (открытой) категории магазина или по всем категориям, если ни одна категория не открыта. Если выбрано несколько категорий, то в форме поиска появится выпадающий список категорий магазина, который будет подгружать прикрепленные к категориям характеристики. Можно указать значение **all**, тогда поиск будет осуществляться по всем категориям товаров и в форме будут участвовать только общие характеристики;

— **ajax** – подгружать результаты поиска без перезагрузки страницы. Результаты подгружаются только если открыта страница со списком товаров, иначе поиск работает обычным образом;

— **template** – шаблон тега (файл *modules/shop/views/shop.view.show_search_template.php*; по умолчанию шаблон *modules/shop/views/shop.view.show_search.php*).

Пример:

```
<insert name="show_search" module="shop">
```

выведет форму поиска по каталогу товаров

show_social_links

Выводит ссылки на социальные сети.

Пример:

```
<insert name="show_social_links">
```

Шаблон: функция `show_social_links` в файле *includes/function.php*.

T

show_tags module="site"

Выводит теги (слова-якори), прикрепленные к странице сайта, если в конфигурации модуля «Страницы сайты» подключены теги.

Атрибуты:

— **template** – шаблон тега (файл *modules/site/views/site.view.show_tags_template.php*; по умолчанию шаблон *modules/site/views/site.view.show_tags.php*).

Пример:

```
<insert name="show_tags" module="site">
```

выведет список ссылок на теги, прикрепленные к текущей странице

show_text

Выводит текст страницы.

Пример:

```
<insert name="show_text">
```

Шаблон: функция `show_text` в файле *includes/function.php*.

show_title

Выводит заголовок. Используется между тегами `<title></title>` в шапке сайта.

Пример:

```
<title><insert name="show_title"></title>
```

выведет заголовок текущей страницы

Шаблон: функция `show_title` в файле *includes/function.php*.

Y

show_year

Выводит период функционирования сайта в годах.

Атрибуты:

— **year** – начало отсчета (по умолчанию текущий год).

Пример:

```
<insert name="show_year">
```

в 2009 году выведет «2009 г.», в 2010 году выведет «2010 г.»

```
<insert name="show_year" year="2010">
```

в 2010 году выведет «2010 г.», в 2013 году выведет «2010-2013 гг.»

Шаблон: функция show_year в файле *includes/function.php*.