

DIAFAN.CMS version 6.0

<http://www.diafan.ru/>



Полное руководство для разработчиков

Содержание

Полное руководство

- [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](#)
 - [Шаблон сайта](#)
 - [Шаблоны модулей](#)
 - [Шаблонные функции и теги](#)
 - [Схема работы шаблонов](#)
 - [Как подключить DIAFAN.CMS к дизайну](#)
- [Системные модули](#)
 - [Страницы сайта](#)
 - [Страницы админки](#)
 - [Меню на сайте](#)
 - [Пользователи сайта](#)
 - [Темы и дизайн](#)
 - [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](#)
- [Ошибки на сайте](#)
- [On-line консультанты](#)
- [Файловый менеджер](#)
- [Личные сообщения](#)
- [Надмодули](#)
 - [Комментарии](#)
 - [Рейтинг](#)
 - [Перелинковка](#)
 - [Панель быстрого редактирования](#)
 - [Оплата](#)
 - [Баланс пользователя](#)
 - [Геокарта](#)
 - [Изображения](#)
 - [Прикрепленные файлы](#)
 - [Постраничная навигация](#)
 - [Карта сайта](#)
 - [Captcha](#)
 - [bbCode](#)
- [Разработчикам](#)
 - [Основные принципы DIAFAN.CMS](#)
 - [Архитектура DIAFAN.CMS](#)
 - [Структура базы данных](#)
 - [Устройство модуля](#)
 - [Контроллер](#)
 - [Модель](#)
 - [Шаблон](#)
 - [Обработка POST-запросов](#)
 - [Подключение](#)
 - [Инсталляция](#)
 - [Прямое обращение к файлам модуля](#)
 - [Взаимодействие с другими модулями](#)
 - [JS-скрипты](#)
 - [CSS-файлы](#)
 - [Подключение модуля](#)
 - [Административная часть сайта](#)
 - [Основные действия](#)
 - [Список](#)
 - [Редактирование](#)
 - [Валидация](#)
 - [Сохранение](#)
 - [Удаление](#)
 - [Публикация/скрытие](#)
 - [Перемещение](#)

- [Обработка POST-запросов](#)
 - [Администрирование модуля](#)
 - [Настройки модуля](#)
 - [Подключение модуля](#)
 - [JS-скрипты](#)
 - [CSS-файлы](#)
- [Подключения \(API\)](#)
 - [База данных](#)
 - [Валидация данных](#)
 - [Изображения](#)
 - [Кастомизация](#)
 - [Конфигурация сайта](#)
 - [Кэширование](#)
 - [Маршрутизатор](#)
 - [Обработка поисковой фразы](#)
 - [Общие функции](#)
 - [Парсер шаблонных тегов](#)
 - [Подключение шаблонов](#)
 - [Почта](#)
 - [Режим разработки](#)
 - [Сессии](#)
 - [Сжатие страниц](#)
 - [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](#)
 - [Изменение административной части](#)
 - [Как в поиске по товарам заменить чекбоксы на список](#)
 - [Как вывести дополнительную характеристику товара в любом месте](#)
 - [Как добавить виджет комментариев из социальной сети](#)
 - [Частичная кастомизация](#)
- [Справочник шаблонных тегов](#)

Наверх

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 уникальный 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 платный хостинг на UNIX-платформе, отвечающий следующим условиям:

- Apache с установленным mod_rewrite
- В настройках Apache должна быть включена директива AllowOverride со значением "All" или "Options FileInfo AutoConfig", при этом директива AccessFileName равна ".htaccess"
- PHP 5.3X и более новые версии (для PHP 5.5 отсутствие OpCache)
- MySQL 5.XX
- Библиотеки GD, Imagick
- Поддержка сокетов и директивы allow_url_fopen
- Для автообновления также желателен --enable-ftp, но это необязательно, если скрипты имеет право записи

[Рекомендуемый хостинг - ISP-Server](#)

[DIAFAN.CMS совместимые хостинги \(список составляется пользователями\)](#)

Установка на localhost

Возможна работа DIAFAN.CMS на локальном хосте. Для этого подойдет любая программная оболочка-эмулятор, то есть набор Apache+PHP+MySQL. Например, [Open Server](#). Локальный хост должен называться "localhost", "diafan", "mysite" или как угодно еще, но без точек в имени (иначе DIAFAN.CMS потребует лицензию). Последовательность установки на локальный сервер точно такая же, как [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. При автономной разработке нужно создать локальную папку. Например, для сайта "hello-world" при типовых настройках Denwer правильный путь будет "c:\WebServers\home\localhost\www\hello-world\". Обратите внимание, что без активированной лицензии могут быть проблемы с локальной работой над сайтами, в названиях которых есть точки. То есть "hello-world" ок, а для "hello-world.com" потребуется лицензия.
2. Наберите в адресной строке браузера адрес своего сайта. При работе на сервере это будет, например, "hello-world.com". В локальном случае - "localhost/hello-world".
3. Затем на экране появятся инструкции установщика DIAFAN.CMS.

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

Установка

Проверка системы

- ✓ Версия PHP > 5.2
- ✓ Поддержка MySQL
- ✓ Поддержка сокетов
- ✓ Папка /userfiles/
- ✓ Папка /cache/
- ✓ Папка /tmp/
- ✓ Папка /custom/
- ✓ Папка /return/
- ✓ Файл install.php
- ✓ Файл config.php
- ✓ Файл index.html

↺ Проверить снова

Далее

Далее следует принять договор-оферту.

Установка

Лицензия

ПОЖАЛУЙСТА, ВНИМАТЕЛЬНО ОЗНАКОМЬТЕСЬ С ТЕКСТОМ ДАННОГО СОГЛАШЕНИЯ ПЕРЕД НАЧАЛОМ УСТАНОВКИ, КОПИРОВАНИЕ, УСТАНОВКА, ЗАПУСК ИЛИ ИНОЕ ИСПОЛЬЗОВАНИЕ ПРОГРАММЫ ЯВЛЯЕТСЯ ПОЛНЫМ И БЕЗОГОВОРЧНЫМ АКЦЕПТОМ (ПРИ ПРИНЯТИИ) ИСПОЛЬЗОВАНИЯ ПРОГРАММЫ ЯВЛЯЕТСЯ НЕДОПУСТИМЫМ.

Содержание

[Термины](#)

[Предмет соглашения](#)

[Авторские права и товарные знаки](#)

[Условия использования программы и ограничения](#)

[Уступка \(передача\) прав](#)

[Техническая поддержка](#)

[Ответственность сторон](#)

[Ограниченная гарантия](#)

[Действие, изменение и расторжение Соглашения](#)

ЛИЦЕНЗИОННОЕ СОГЛАШЕНИЕ НА ИСПОЛЬЗОВАНИЕ ПРОГРАММЫ ДЛЯ ЭВМ «Система управле

Настоящее Лицензионное соглашение (далее – «Соглашение») заключается между ООО «Диафан» (далее – «Лицензиар») и любым неотъемлемой частью Правил оказания информационных услуг, что опубликованы в сети «Интернет» по адресу: <http://user.diafan.ru> документами (п. 2.4 Правил).

Установка, запуск или иное использование Программы разрешены исключительно после акцепта (принятия) Пользователем положения <http://www.diafan.ru/kupit/>, <http://cloud.diafan.ru/tarif/>, <http://addons.diafan.ru/>.

1. Термины

1.1. В настоящем Соглашении, если из текста прямо не вытекает иное, следующие слова и выражения имеют указанные ниже значе

1.1.1. Система управления сайтами DIAFAN.CMS (далее – «Программа») – программа для ЭВМ предназначенная для управления Сайтом, команд, в том числе исходного текста, базы данных, документация по ее использованию, аудиовизуальных произведений, включенн

Назад

Принимаю

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

Создать базу данных проще всего с помощью утилиты phpMyAdmin. Она входит в состав "джентельменского набора" Denver. В интернете множество пошаговых описаний действий для создания и настройки баз данных MySQL. Не забудьте создать пользователя с полными правами на все операции с БД - эти данные понадобятся на следующем шаге установки DIAFAN.CMS.

Установка

Параметры

Название сайта

Морской берег

База данных MySQL (кодировка базы должна быть UTF-8)

Host

localhost

База данных

dbase

Пользователь

root

Пароль

password

Префикс таблиц

prefix_

☐

Перезаписать

существующие таблицы

Назад

Далее

Далее следует создать главного администратора системы. Нужно заполнить все поля. Старайтесь не использовать слишком простые пароли (и конечно не повторяйте здесь пароль к базе данных сайта).

Установка

Администратор сайта

ФИО

Иванов Иван

Логин

login

Пароль

password

E-mail адрес

mail@mail.ru

Папка административной части

admin

Папка для хранения пользовательских файлов

userfiles

Назад

Далее

Затем необходимо указать дополнительные модули, которые необходимо установить.

На этом этапе будет предложено заполнить сайт примерами из демо-версии «Магазин туристического оборудования». Это удобно, когда нужно быстро получить детальный пример сайта. Но если изначально задача в создании нового ресурса с уникальным контентом, не нужно ставить галочку на заполнении сайта - иначе потом придется выполнять лишние действия по удалению автоматически созданных примеров.

Внимание! На этом шаге можно указать файл-архив с тематическим шаблоном, скачанный на сайте addons.diafan.ru. Тогда все параметры выше будут проигнорированы, а на Ваш сайт будет установлен предзаполненный тематический сайт с проекта «Дополнения».

Установка

Конфигурация

Установить базовую версию

- | | | | |
|---|---|--|--|
| <input checked="" type="checkbox"/> Интернет-магазин | <input checked="" type="checkbox"/> Новости | <input checked="" type="checkbox"/> Статьи | <input checked="" type="checkbox"/> Фотогалерея |
| <input checked="" type="checkbox"/> Теги | <input checked="" type="checkbox"/> Баланс пользователя | <input checked="" type="checkbox"/> Оплата | <input checked="" type="checkbox"/> Форум |
| <input checked="" type="checkbox"/> Рассылки | <input checked="" type="checkbox"/> Рейтинг | <input checked="" type="checkbox"/> Комментарии | <input checked="" type="checkbox"/> Ошибки на са |
| <input checked="" type="checkbox"/> Файловый менеджер | <input checked="" type="checkbox"/> Поиск | <input checked="" type="checkbox"/> Личные сообщения | <input checked="" type="checkbox"/> Карта сайта |

☐ Заполнить сайт демо-контентом (займет несколько минут)

☐ Создать две языковые версии сайта (Русский/English)

или

Установить тематический сайт

[Загрузить шаблон сайта](#)

(выбрать шаблон тематического сайта можно в [каталоге шаблонов](#))

[Назад](#)

[Далее](#)

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

Установка

Окончание

 **DIAFAN.CMS успешно установлена**

[Перейти на сайт](#)

[Открыть панель управления](#) (логин: login пароль: password).

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 тестовом режиме 21 день, не требуя оплаты. Когда данный период закончится, выйдет соответствующее уведомление. Далее необходимо либо прекратить использование системы, либо 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 создавать на DIAFAN.CMS множество сайтов, просто приобретая лицензии и добавляя имена доменов в личном кабинете.

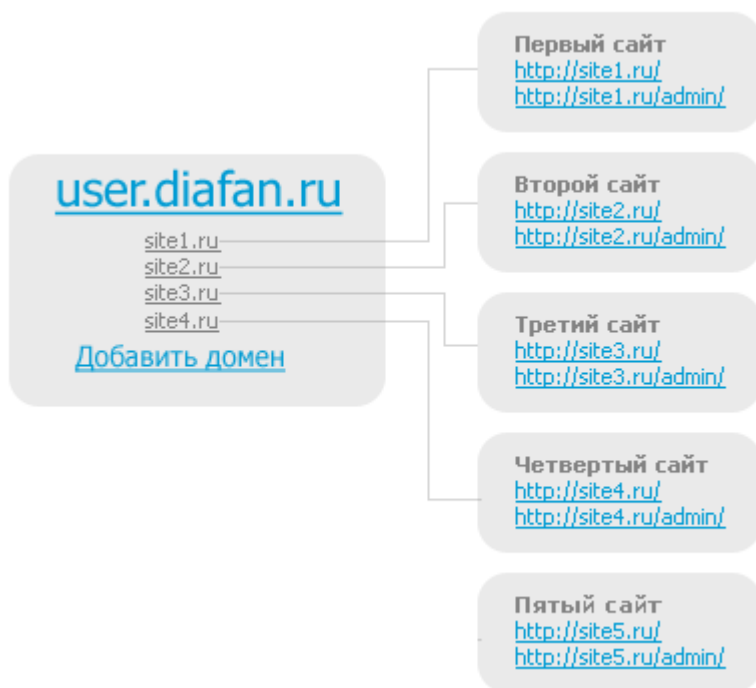


Лицензия на основной домен действует также и для всех его поддоменов, то есть если вы купили лицензию для адреса site.ru, вы сможете использовать DIAFAN.CMS и на сайте с адресом shop.site.ru и company.site.ru.

Для тестирования и разработки на локальном веб-сервере лицензия не требуется — создайте для этого локальный адрес без точки — localhost, diafan, mysite и т.п. Важно помнить, что на адресах с точкой, diafan.loc или mysite.com система запросит лицензию. Также автоматическое обновление системы для локальных сайтов невозможно.

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

Схематично связь между личным кабинетом, лицензиями в нем и сайтами можно изобразить так:



На каждый из четырех первых сайтов лицензии есть, поэтому на них система будет работать. А вот административная часть пятого сайта без лицензии в ЛК, поэтому спустя 21 день после установки она работать перестанет. При попытке перейти по адресу <http://site5.ru/admin/> будет выведено соответствующее сообщение. Продолжать использовать DIAFAN.CMS в таком режиме запрещено.

Использование DIAFAN.CMS на сайте без лицензии является противоправным действием и уголовно наказывается согласно ст. 146 Уголовного кодекса Российской Федерации «Незаконное использование объектов авторского права». Кроме того, нарушение исключительных имущественных прав влечет гражданско-правовые санкции согласно ст. 1301 Гражданского Кодекса Российской Федерации в виде возмещения причиненных правообладателю убытков путем выплаты ему компенсации в размере от 10 тысяч до 5 миллионов рублей либо в двукратном размере стоимости экземпляров произведения по собственному выбору. Владельцу сайта будет направлена претензия. Также, соответствующая претензия будет направлена в адрес хостинг-провайдера сайта и регистратора используемого доменного имени.

Файловая структура DIAFAN.CMS

- [Что есть что?](#)

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

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

Что есть что?

adm – папка панели администрирования. Прямое доступа *http://site.ru/adm/* при запросе из адресной строки не имеет. На сайте может иметь любой адрес (по умолчанию *http://site.ru/admin/*), задаваемый в процессе инсталляции DIAFAN.CMS в любом виде, например, *http://site.ru/papka-admina/*. После инсталляции этот адрес из настроек сайта можно изменить. В папке хранится ядро системы управления сайтом, оформление, картинки панели управления, визуальный редактор и вспомогательные функции для управления сайтом. Как правило, менять и править веб-мастеру там нечего.

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

css – папка с CSS-стилями оформления [стандартного шаблона дизайна](#), поставляемого с

DIAFAN.CMS, CSS-стили для jQuery-плагинов.

custom – [пользовательские измененные модули и шаблоны дизайна](#), необходимо установить на папку **custom** права **777**.

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.

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

langs – файлы перевода интерфейса.

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

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

return – архив [точек возврата](#) (резервных копий предыдущих обновлений), необходимо установить на папку **return** права **777**.

themes – папка [стандартного шаблона дизайна](#). [Основной шаблон](#) дизайна сайта – файл *site.php*. Подробнее о шаблонах дизайна можно прочитать [здесь](#).

tmp – папка для временных файлов. Чтобы DIAFAN.CMS успешно работала, необходимо установить на папку **tmp** права **777**.

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 (рекурсивно) права **777**. Имя папки, по которому будут доступны все файлы пользователям сайта, можно изменить в настройках сайта.

.htaccess – управляющий файл DIAFAN.CMS. Достаточно важный файл. Обеспечивает работу псевдоссылок, кодировки, переадресаций и пр. В случае изменений данного файла работоспособность DIAFAN.CMS не гарантируется.

index.html – файл кэша главной страницы. Раз в сутки в этом файле создается полная копия внешнего вида и контента главной страницы. Если на хостинге происходит сбой и перестает работать БД или интерпретатор, система открывает на сайте содержимое данного файла. Таким образом, главная страница сайта не бывает «На техобслуживании» из-за сбоя в БД, что особенно важно, когда сайт SEO-продвигается. Необходимо установить на этот файл права **777**.

index.php – инициализирующий файл DIAFAN.CMS, стартовый. Изменений не требует.

install.php – файл установки DIAFAN.CMS. После установки системы удаляется. Необходимо установить на этот файл права **777**.

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 установить на этот файл права **777**.

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

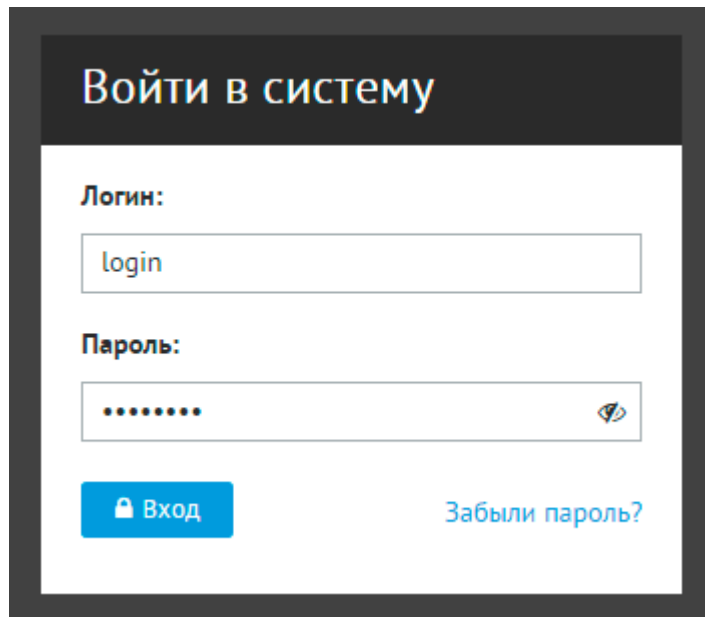
update.txt – руководство по обновлению DIAFAN.CMS с предыдущей версии.

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:




Войти в систему

Логин:

Пароль:





 Вход


[Забыли пароль?](#)

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


Контент


 Страницы сайта


 Меню на сайте


 Новости

 Статьи

 Фотогалерея


 Баннеры


 Файловый архив


 Объявления 3/0


 Теги


Интернет магазин

 Каталог


 Заказы 2


 Справочники


 Оплата


 Статистика


Интерактив


 Пользователи сайта


 Форум

 Вопрос-Ответ 1

 Обратная связь 2

 Опросы

 Рассылки

 Геокарта

 События

Страницы сайта

Страницы сайта

Блоки на сайте

Динамические блоки

 Добавить страницу сайта



Выбрать всё

Сделать активным



Применить

Сортировка

Название

 Главная страница



1

 О компании



2

 Полезное



3

 Наши магазины




5

 Страница пользователя Подключен модуль:  (



14

Обратная связь Подключен модуль:  Обратная



17

Карта сайта Подключен модуль:  Карта сайта



21

Поиск Подключен модуль:  Поиск



22

 Интернет-магазин Подключен модуль:  Интер



26

Рассылки Подключен модуль:  Рассылки



27

Теги Подключен модуль:  Теги



Выбрать всё

Сделать активным



Применить

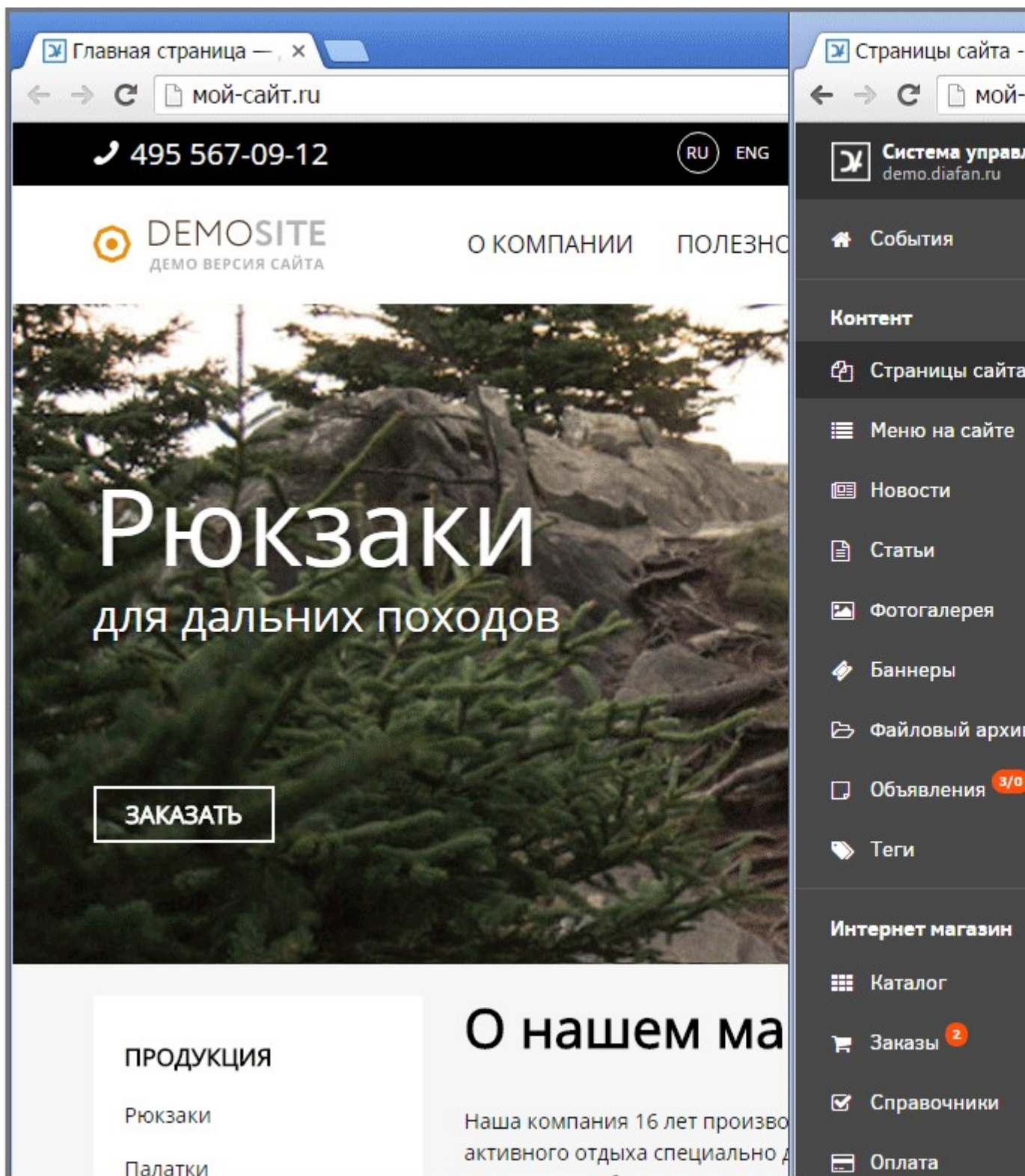
Первое знакомство с DIAFAN.CMS

- [Панель администрирования на Сайте](#)
- [Админка](#)
- [Модули DIAFAN.CMS](#)

Вы стали счастливым обладателем сайта на DIAFAN.CMS, поздравляем!

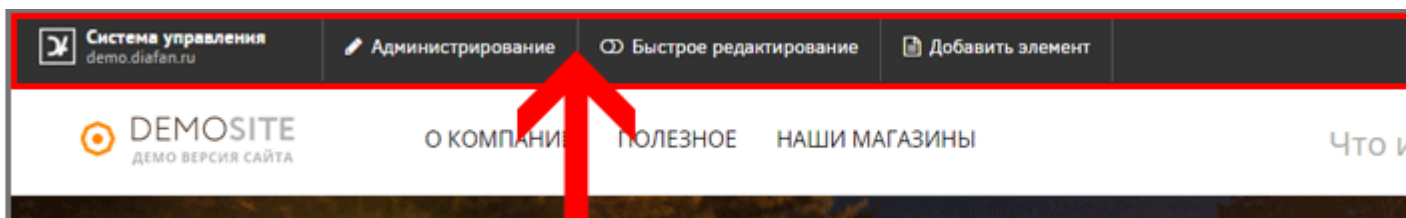
Основные вводные понятия:

- **Пользовательская часть сайта** (далее просто **Сайт**) - доступна по адресу www.мой-сайт.ru всем пользователям интернета. Это и есть Сайт, в обычном понимании этого слова.
- **Административная часть сайта** (далее просто **Админка**) - доступна по секретному адресу, известному только Вам, вида www.мой-сайт.ru/секретный-адрес/, требующая ввода логина и пароля администратора Сайта. Служит для управления Сайтом.

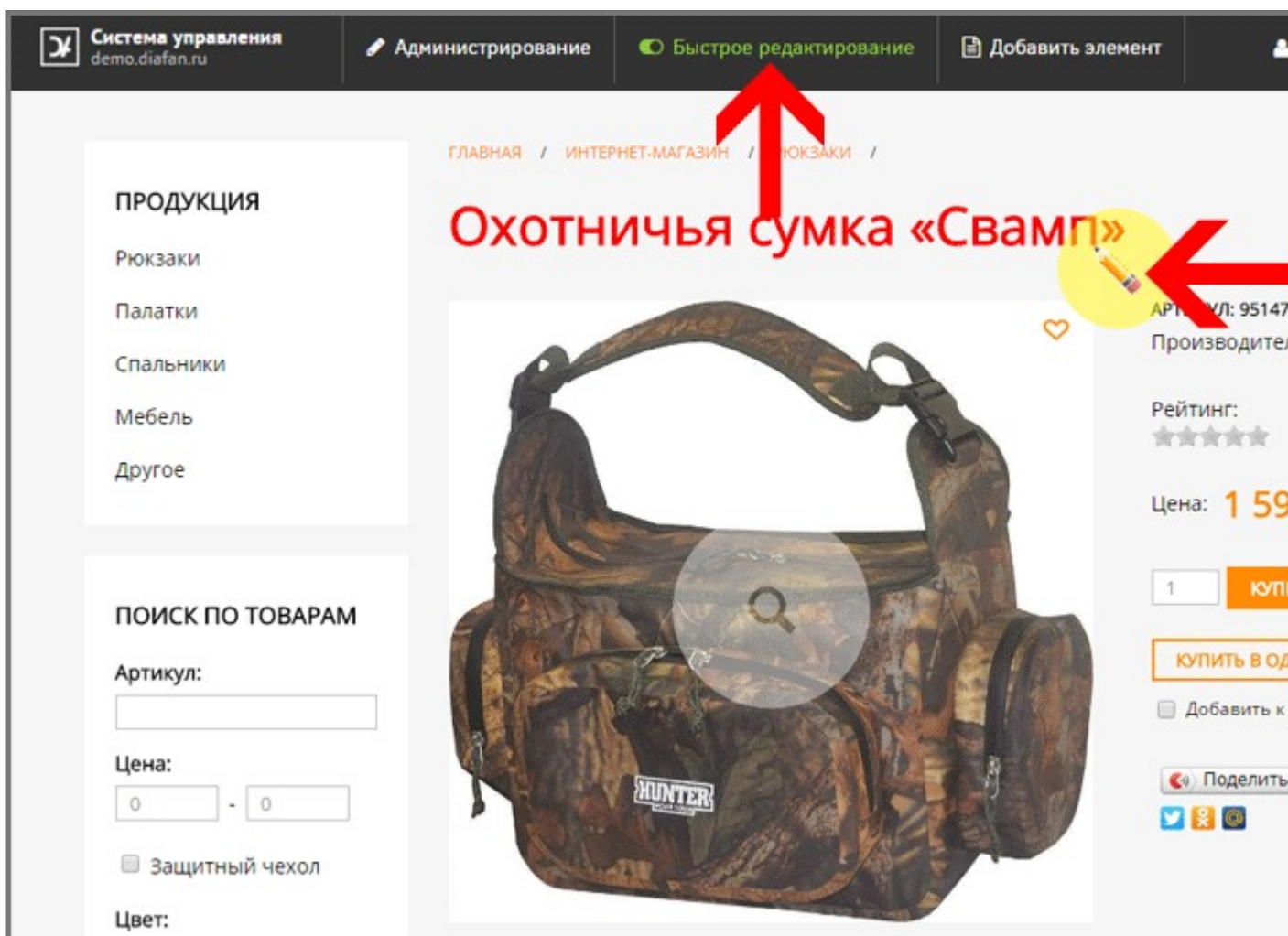


Панель администрирования на Сайте

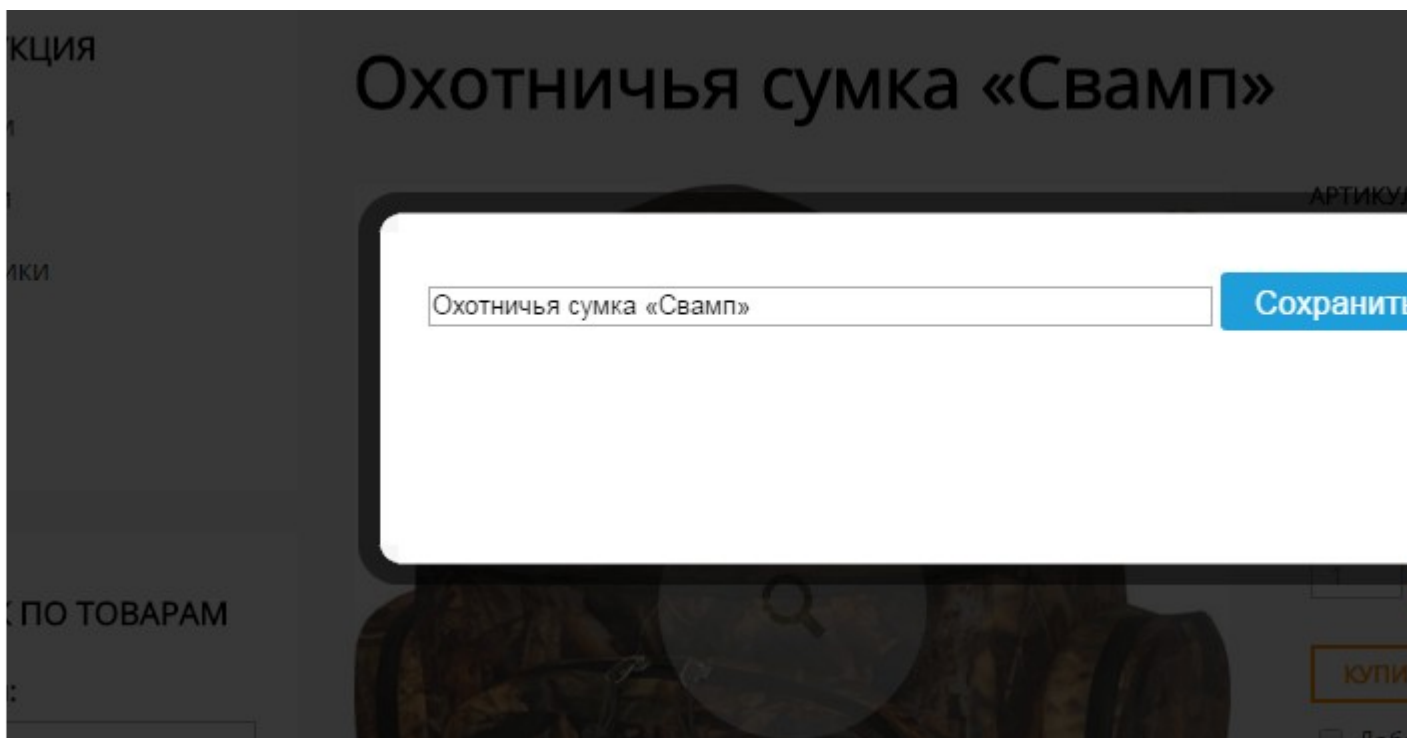
После того, как Вы авторизуетесь в Админке, на Сайте появится панель.



Она доступна только Вам, обычные посетители её не видят. На панели дублируются некоторые возможности из Админки, однако её главная возможность - «Быстрое редактирование». По клику на нее включается режим редактирования той страницы сайта, на которой Вы находитесь. Подводите курсор мышки к любым* текстам на Сайте, и если есть возможность изменить этот текст, курсор превратится в карандаш, а текст подсветится красным.



Кликните для правки.



Менять можно почти всё*: тексты страниц, блоки, номера телефонов, заголовки, названия товаров, цены, пункты меню и ссылок. После того, как Вы внесете изменения, не забудьте отключить «Быстрое редактирование», чтобы иметь возможность перейти на другую страницу Сайта.

Если в панели нажать на соседнюю кнопку «Администрирование», текущая страница (категория, новость или товар) откроется для полного редактирования в Админке.

Админка

Для управления сайтом служат разнообразные модули, их список в Админке слева. Каждый модуль заточен под удобное управление тем или иным содержимым сайта.

Страницы сайта

Основной модуль системы - Страницы сайта, служит для управления основной структурой сайта, и создания обычных текстовых страниц сайта. Как и все модули, Страницы сайта расположен на панели модулей слева. Как и во многих других модулях, у Страниц сайта есть свои вкладки, «Блоки на сайте» и «Динамические блоки» являются частью модуля.

Система управления
demo.diafan.ru

События

Контент

Страницы сайта

Меню на сайте

Новости

Статьи

Фотогалерея

Баннеры

Файловый архив

Объявления 3/0

Теги

Интернет магазин

Каталог

Заказы 2

Справочники

Оплата

Статистика

Интерактив

Пользователи сайта

Форум

Вопрос-Ответ 1

Обратная связь

Просмотр сайта

Добавить элемент

rueng

События

Страницы сайта

Блоки на сайте

Динамические блоки

Добавить страницу сайта

Показы

Выбрать все

Сделать активным

Применить

Сортировка

Название

Главная страница

1

О компании

2

Полезное

3

Наши магазины

5

Страница пользователя

Подключен модуль: Страница пользователя

14

Обратная связь

Подключен модуль: Обратная связь

17

Карта сайта

Подключен модуль: Карта сайта

21

Поиск

Подключен модуль: Поиск

22

Интернет-магазин

Подключен модуль: Интернет-магазин

26

Рассылки

Подключен модуль: Рассылки

27

Теги

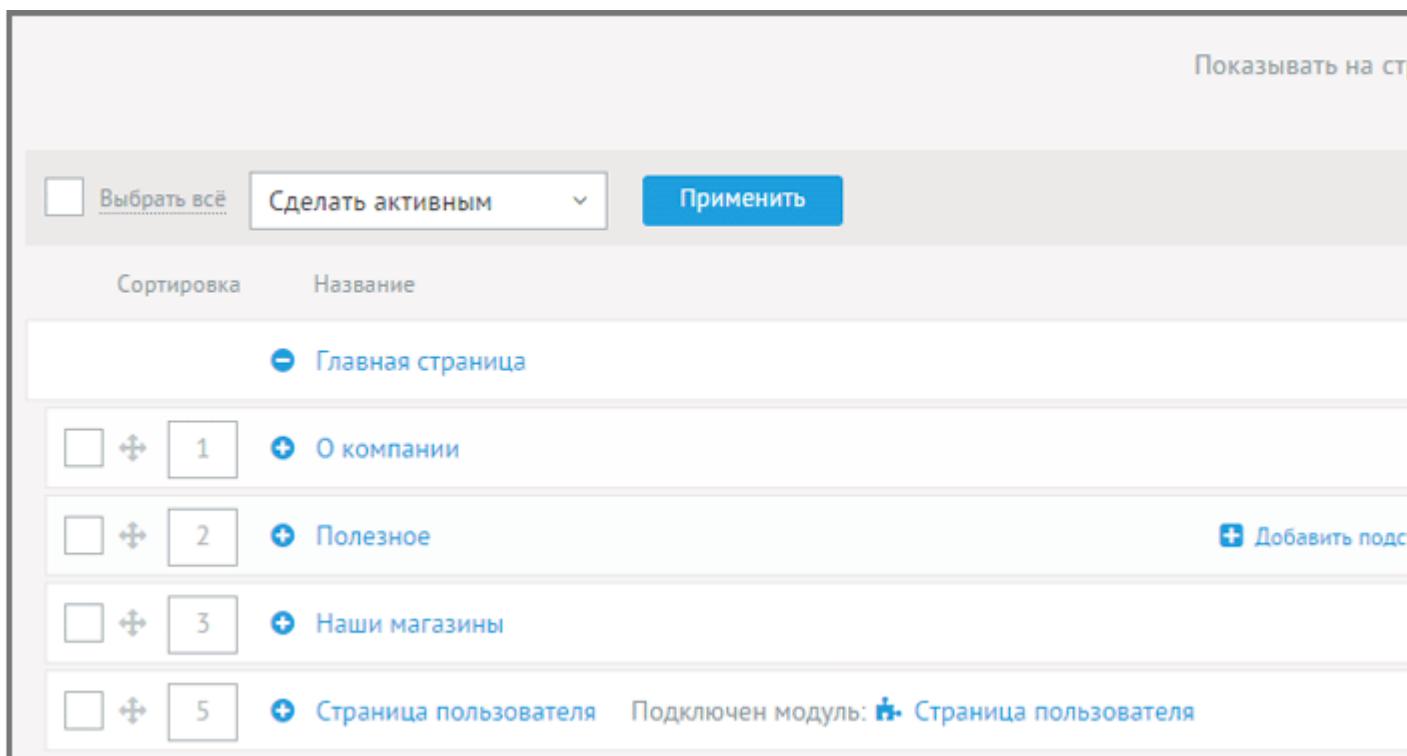
Подключен модуль: Теги

Выбрать все

Сделать активным

Применить

Как и все остальные модули, Страницы сайта выводят информацию в виде списка.



У всех списков есть служебные кнопки для управления списком.

Сортировка - первое поле в списке, сортировать элементы можно двумя способами, перетаскиванием мышкой («хватать» нужно за крестик), или введя сортировочный индекс в поле вручную (чем больше, тем ниже).

Плюсик слева от названия кликабелен, означает, что у элемента есть вложенность и его можно «раскрыть».

Клик по названию переведет в редактирование элемента списка.

Добавить подстраницу создает подстраницу в выбранном элементе списка. Создав подстраницу, слева от названия появится тот самый «Плюсик» +.

Посмотреть на сайте - элемент откроется в отдельном окне на Сайте в том виде, как его увидят посетители сайта. Даже если элемент неактивен, Вы, как администратор, всё равно его увидите - это позволяет готовить страницы, просматривая результат, без публикации на Сайте.

Сделать активным / неактивным. Нажатие на эту кнопку позволяет сделать неактивным (невидимым для посетителей, кроме администратора) на сайте.

Удалить удаляет элемент и все его вложенности, если таковые имелись, в корзину.

Нажмите «Добавить страницу сайта» над списком, чтобы добавить страницу.

Для создания страницы достаточно заполнить её название, и вписать текст. Адрес страницы, её заголовок сформируются автоматически, но их всегда можно изменить вручную в разделе «Дополнительные параметры».

Страницы сайта

Страницы сайта

Блоки на сайте

Динамические блоки

Добавить новый

Основная информация

Дополнительные

Название ?

Название новой страницы

☒ Опубликовать на сайте ?

Контент страницы ?

Изменить ▾ Вставить ▾ Таблица ▾

<> **B** *I* U ~~S~~ x^2 x_2 A ▾ **A** ▾

Параграф ▾ Шрифт ▾ Размер шрифта ▾

Текст, который будет на новой странице Вашего сайта|

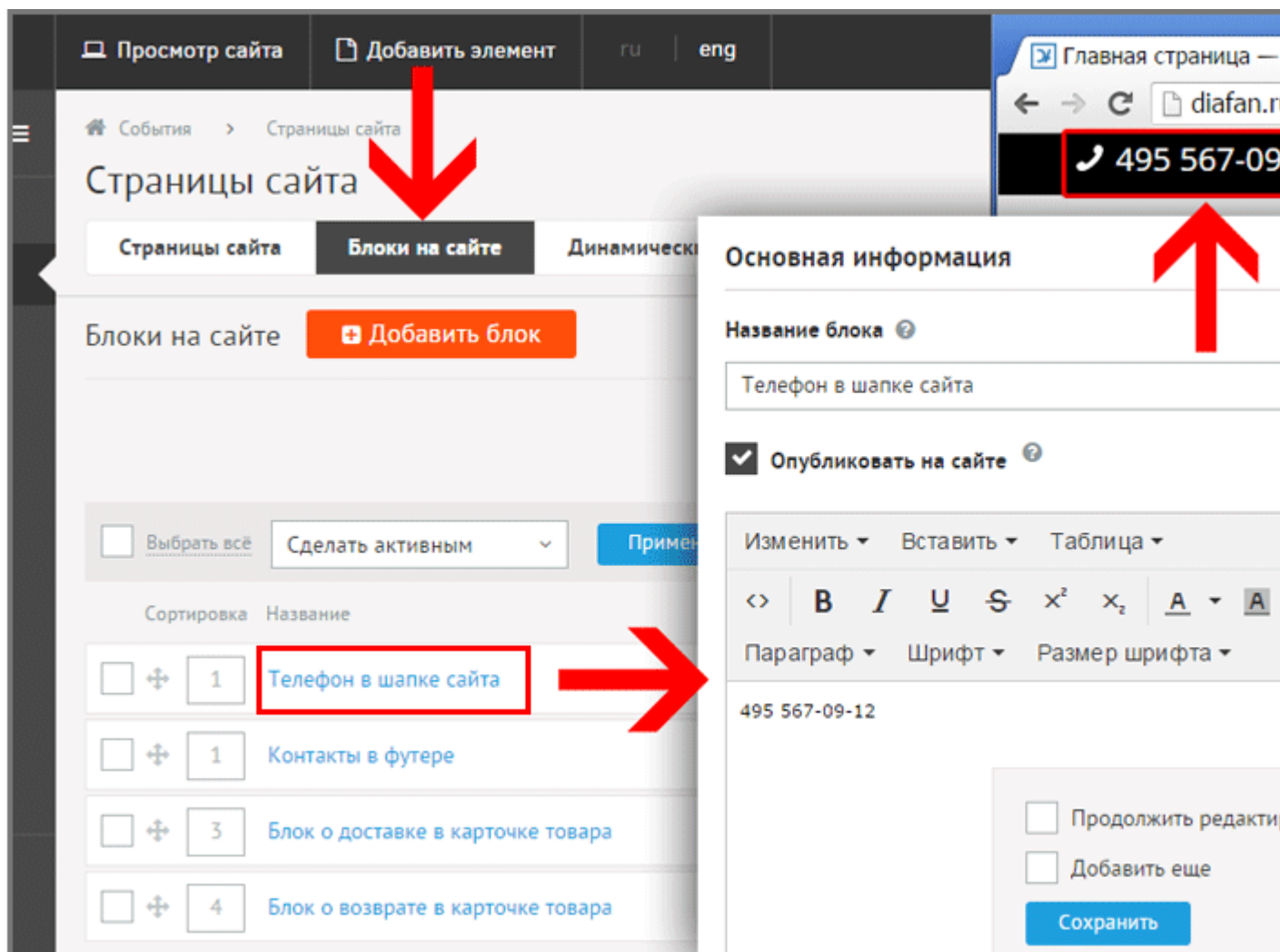
☐ Продолжить редактирование

☐ Добавить еще

Сохранить

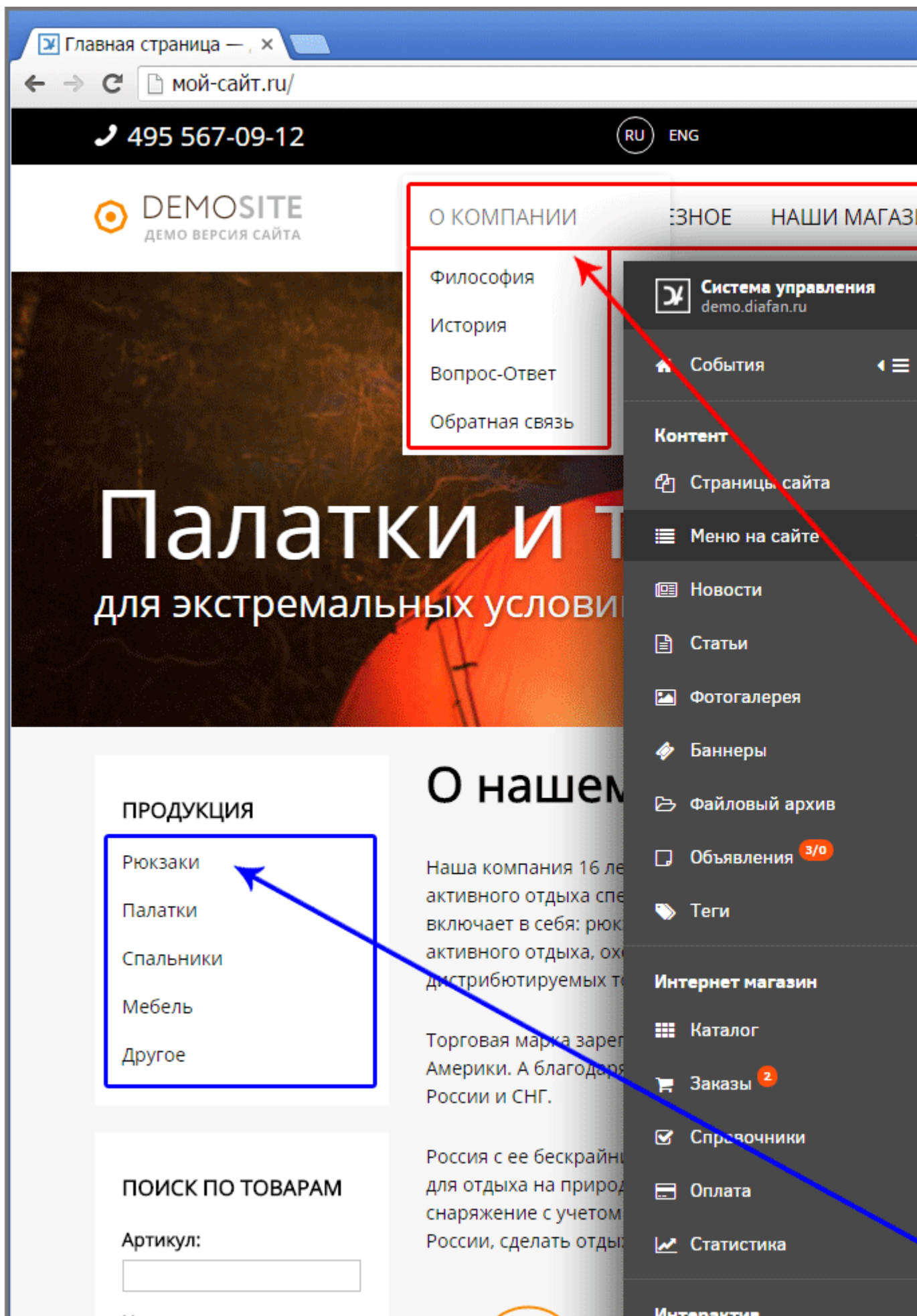
Модуль **«Страницы сайта»** - единственный модуль, который работает самостоятельно и не зависит от других модулей. Напротив, все остальные модули так или иначе зависят от страниц сайта, или работают как его части. Об этом чуть ниже.

Вкладка **«Блоки на сайте»** служит для управления блоками. Это какие-то участки сайта, вынесенные веб-мастером для правки в Админку. Яркий пример - номера телефонов.

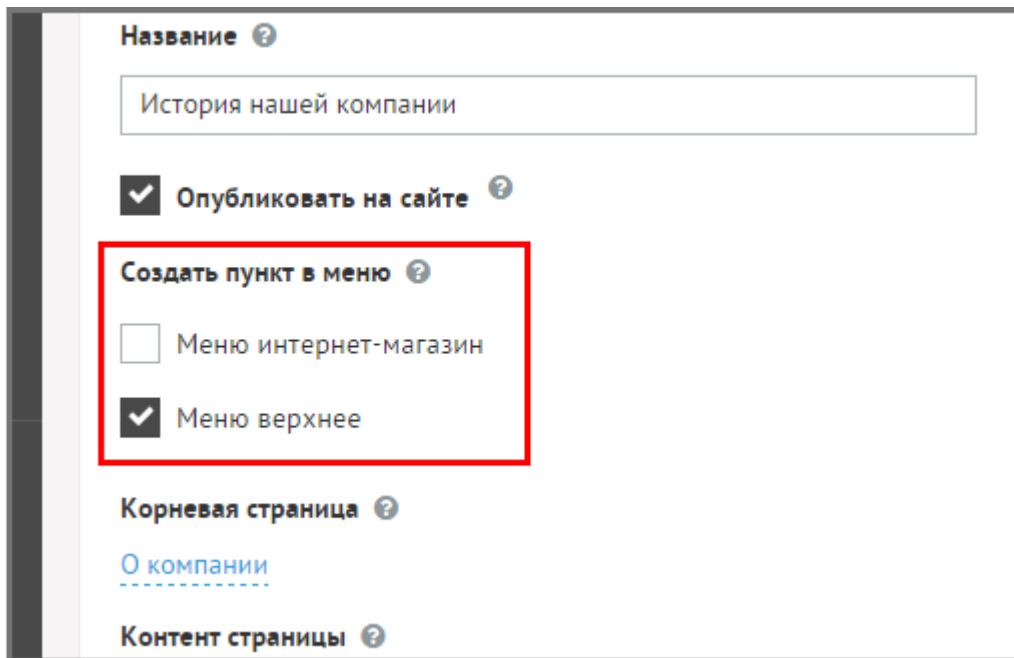


Меню на сайте

Еще один важный модуль, работающий вместе со страницами сайта - **«Меню»**. Меню - это блок на Сайте, выводящий ссылки на страницы сайта. На Сайте может быть несколько меню, например «верхнее меню», «нижнее меню», «меню с услугами», «каталог товаров» и т.д. Всё зависит от дизайна Вашего Сайта. Редактировать пункты меню нужно именно в этом модуле.



Для Вашего удобства при создании страницы сайта в модуле «Страницы сайта» (или категории в модуле «Интернет-магазин», или альбома в модуле «Фотогалерея») можно отметить галку «Создать пункт меню» и тогда в модуле «Меню» автоматически будет создан пункт.



The screenshot shows a web application interface for creating a page. It includes a text input field for the page name, a checkbox for publishing on the site, and a section for creating a menu item. The 'Create menu item' section is highlighted with a red box and contains two options: 'Menu internet-store' (unchecked) and 'Menu top' (checked). Below this section are links for the root page and page content.

Название ?

История нашей компании

☒ Опубликовать на сайте ?

Создать пункт в меню ?

☐ Меню интернет-магазин

☒ Меню верхнее

Корневая страница ?

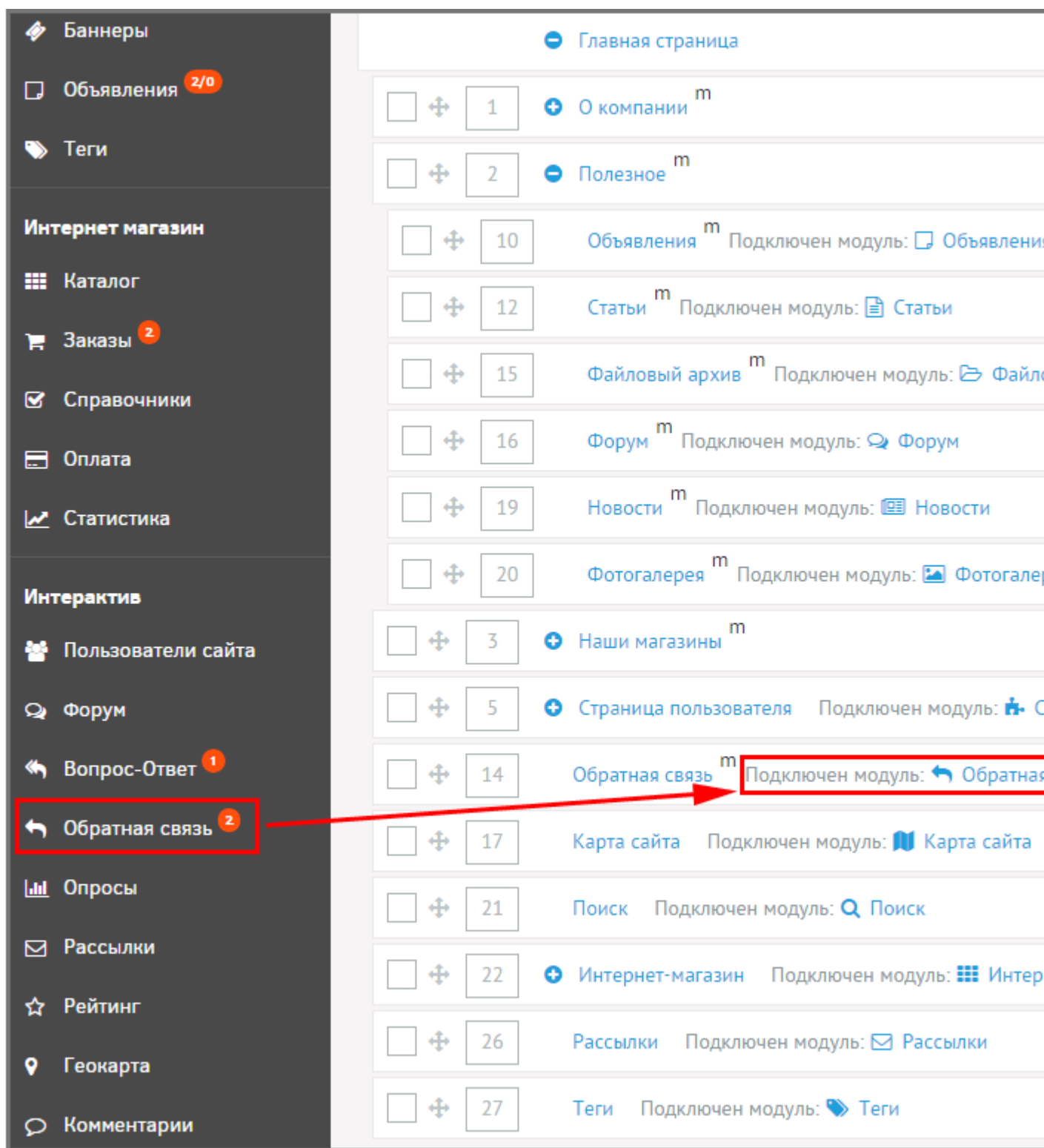
[О компании](#)

Контент страницы ?

Модули DIAFAN.CMS

Все модули организованы по одному и тому же принципу: «Список»-«Элемент». Ни один модуль не может работать сам по себе, он обязательно должен быть прикреплен к какой-нибудь странице сайта.

На иллюстрации ниже существующие в системе модули подключены к разным страницам сайта. Перейти в редактирование элементов модуля можно с панели слева или из страниц сайта, кликнув на «Подключен модуль». Оранжевый информер возле названия модуля на панели слева означает, что в модуле произошло событие, например, появились новые сообщения от посетителей сайта.

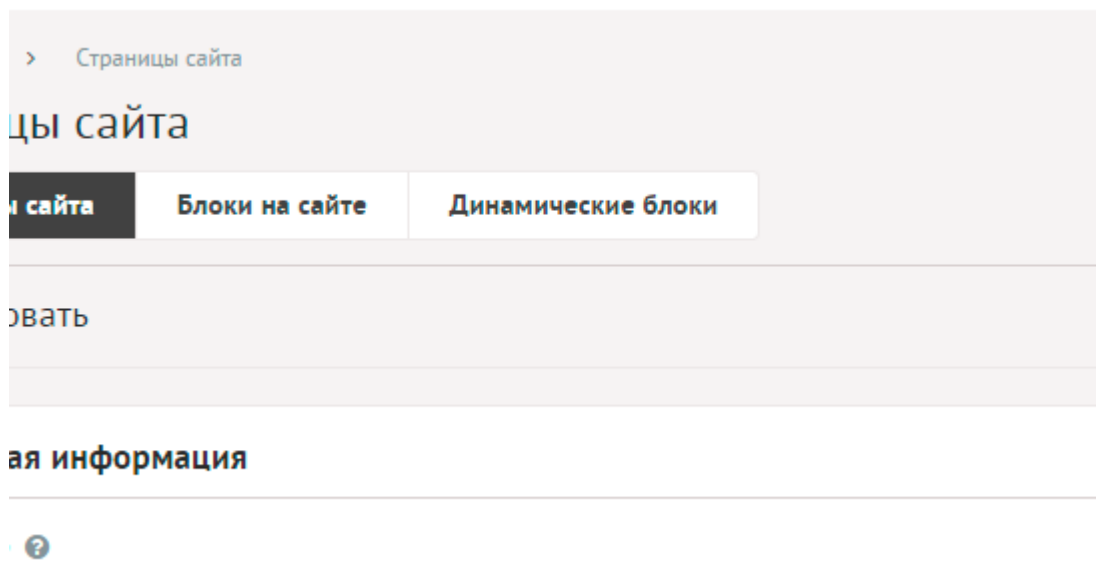


На странице сайта «Новости» с адресом www.ваш-сайт.ru/news/ прикреплен модуль «Новости». Соответственно, все элементы модуля (новости) будут размещаться вглубь адреса **www.ваш-сайт.ru/news/**: www.ваш-сайт.ru/news/novost1/ www.ваш-сайт.ru/news/novost-vtoraya/ и т.д.

Внимание! [Смотрите видеоурок на тему первого знакомства с DIAFAN.CMS](#) 20 минут просмотра и Вы будете иметь общее представление относительно базовых возможностей DIAFAN.CMS.

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:



☐ HTML-код

HTML-код. Отображает содержимое (HTML код) страницы без визуального редактора.

☐ Применить типограф

Применить типограф. Применение экранной типографики при вёрстке материалов.

Изменить ▾

Изменить. Позволяет вернуть или отменить действие; вырезать, копировать, вставить или вставить как текст; выделить всё.

Вставить ▾

Вставить. Позволяет вставить/редактировать ссылку, видео, изображение или добавить горизонтальную линию.

Таблица ▾

Таблица. Позволяет вставить и редактировать таблицу, форматировать ячейку, столбец или строку таблицы.

<>


Редактировать HTML-код. Отображает HTML код страницы.


A


Замена.


B


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).


 **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).


 **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.


 **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.

 **Заголовок 1** ▾ **Формат.**

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

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

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");

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

//версия DIAFAN.CMS
define("VERSION_CMS", "5.4");

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

//мобильная версия true/false (да/нет)
define("MOBILE_VERSION", true);

//demo-версия true/false (да/нет)
define("IS_DEMO", false);

//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_username");

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

//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_username");

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

//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);
```

//подключить SMS-уведомления true/false (да/нет)

```
define("SMS", false);
```

// ключ для сервиса byteHand

```
define("SMS_KEY", "");
```

// id в системе byteHand

```
define("SMS_ID", "");
```

// подпись для уведомлений

```
define("SMS_SIGNATURE", "1.10.2014 13:21");
```

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

```
define("LAST_1C_EXPORT", "");
```

// разрешить вставлять во frame

```
define("NO_X_FRAME", false);
```

Интеграция дизайна, шаблоны и шаблонные функции

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

Шаблоны

Для начала теория и определения.

Шаблоны – это файлы оформления сайта. Бывают двух видов, **шаблоны сайта**, задающие основное оформление и структуру сайта, и **шаблоны модулей**, которые оформляют внешний вид модулей.

[Шаблон\(ы\) сайта](#) – это оформление страниц(ы) сайта в пользовательской части. Это и есть дизайн всего сайта. Всё то, что видят посетители сайта: логотип, шапку сайта, блоки, контентную зону, подвал. По умолчанию DIAFAN.CMS поставляется с базовым шаблоном дизайна (так выглядит [демо-версия](#)) и его файлы хранятся в папке *themes*.

[Шаблоны модулей](#) – это файлы оформления модулей. Находятся в *modules/имя_модуля/views/имя_модуля.view.название_шаблона.php*. Шаблоны модулей отвечают за оформление модульной информации в шаблонах сайта: список новостей, товаров, статей, карточки товаров, голосования и т.д. Если **шаблоны сайта** оформляют шапку сайта, и подвал, которые везде одинаковые. То **шаблоны модулей** оформляют содержимое.

Шаблонные функции и теги

Шаблонные функции – это функции для вывода контента модулей 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 файлы из папки *themes/functions* общего назначения;
- [шаблонные функции модулей](#) – это функции, служащие для вывода на сайте модульной информации: контента, заголовков, блоков, списков новостей, статей, товаров, фильтра-поиска по товарам, голосований и т.д.

Шаблон сайта

- [Мультидизайн страниц сайта](#)
- [Мультидизайн сайта](#)

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

Основной шаблон сайта носит имя – *site.php*. Стандартный дизайн DIAFAN.CMS находится в шаблоне *themes/site.php*, однако если на сайте используются [темы](#), аналогичный файл (файлы) может находиться в **custom/имя-темы/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.

В шаблон можно добавлять PHP-код, выделив его стандартными тегами `<?php *** ?>`.

Внимание! При внедрении собственного дизайна быстрее и проще всего взять исходный файл, входящий в пакет DIAFAN.CMS по умолчанию *themes/site.php* и заменить HTML-код, находящийся между тегами: `<body></body>` на свой. Затем останется просто [расставить шаблонные теги](#).

Мультидизайн страниц сайта

DIAFAN.CMS поддерживает мультидизайн страниц 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 *onlynews.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.

Мультидизайн сайта

DIAFAN.CMS поддерживает мультидизайн сайта. Это означает, что внешний вид всей пользовательской части сайта можно менять в один клик.

Для этого служит модуль [темы](#)

Шаблоны модулей

Шаблон модуля – это совокупность файлов в папке *modules/имя_модуля/views*.

Шаблоны модулей отвечают за оформление вывода модульной информации в основной контентной зоне сайта. Это списки новостей, товаров, файлов, фотографий, а также отдельные страницы статей, карточки товаров, блоки голосований, формы отзывов и т.д.

Обычно состав файлов шаблона модуля следующий:

- *модуль.view.first_page.php* – главная (первая) страница модуля;
- *модуль.view.list.php* – список элементов модуля.
- *модуль.view.id.php* – страница отдельного элемента.
- *модуль.view.form.php* – форма для заполнения (если модуль интерактивный).
- *модуль.view.show_block.php* – блок.

Где *модуль* – это название модуля.

Пример:

Шаблон первой страницы каталога интернет-магазина будет в файле *modules/shop/views/shop.view.first_page.php*, а список новостей – *modules/news/views/news.view.list.php*.

Оформление внешнего вида карточки товара в файле *modules/shop/views/shop.view.id.php*.

А оформление формы обратной связи будет в шаблоне *modules/feedback/views/feedback.view.form.php*.

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

Пример:

Если необходимо изменить оформление блока опросов на сайте, то нужно вносить изменения в шаблон модуля опросы: *modules/votes/views/votes.view.show_block.php*

Подключение шаблона происходит следующим образом:

Пример:

```
echo $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:

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

В шаблоне модуля можно вызвать шаблонный тег. Для этого его нужно обработать функцией `htmleditor()`.

Пример:

```
echo $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:

```
// в файле modules/news/news.model.php  
$row["text"] = $this->diafan->_tpl->htmleditor($row["text"]);
```


Шаблонные функции и теги

- [Шаблонные теги](#)

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=""]>
```

Используются шаблонные теги в шаблонах сайта при [подключении дизайна к DIAFAN.CMS](#).

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

Определенные атрибуты:

- **name** – задает вызываемую функцию или выводимую переменную;
- **module** – задает название модуля, к которому относится вызываемая функция.
- **value** – значение выводится на странице. Можно изменять значение в интерфейсе [«Языки сайта» – «Перевод интерфейса»](#) или с помощью [панели быстрого редактирования](#);

В шаблонном теге обязательно должен быть один из атрибутов: **name** или **value**.

Пример:

```
<insert name="show_body">
```

Выведет основной контент сайта: заголовок H1, текст и содержимое модуля, если он подключен к странице.

Пример:

```
<insert name="show_block" module="site" id="1">
```

Выведет содержимое блока ID=1 из модуля «Страницы сайта».

Пример:

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

Выведет форму авторизации для пользователей сайта

Пример:

```
<insert name="show_block" module="menu" id="2" template="leftmenu">
```

Выведет меню ID=2. Шаблон оформления – файл
modules/menu/views/menu.view.show_block_leftmenu.php

Шаблонный тег также можно вызвать из [шаблонов модулей](#). Для этого его нужно обработать функцией `htmleditor()`.

Пример:

```
echo $this->htmleditor('<insert name="show_block" module="shop" sort="sale">');
```

При использовании в шаблоне */modules/news/views/news.view.id.php* выведет на странице с отдельной новостью три товара, лидеров продаж.

Все допустимые значения определенных атрибутов **name** и **module** собраны в [справочник шаблонов тегов](#).

Определенный атрибут **value** служит для автоматического перевода текстов в мультиязычных сайтах (`value="значение"`);

Пример:

```
<insert value="Привет, мир!">
```

Выведет просто фразу «Привет, мир!» 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](#).

Если в административной части существует перевод для фразы «Привет, мир!» – «Hello, world!», то в пользовательской части сайта этот же тег выведет «Hello, world!» при переключении на английскую версию.

Неопределенные атрибуты – это все остальные атрибуты, которые могут быть использованы в шаблонном теге. Они передаются функции в качестве массива в формате:

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

Значения неопределенных атрибутов зависят от модуля, к которому они относятся. Согласно синтаксису значения неопределенных 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 ```

Пример:

```
<insert name="show_block" module="menu" id="3" tag_start_1="[li  
class="bottom"]" tag_start_2="[/li]">
```

Выведет меню ID=3, в котором каждый пункт меню будет оформлен в конструкцию <li class="bottom">пункт меню.

[Справочник шаблонных тегов](#)

Как работают шаблоны сайта и модулей в DIAFAN.CMS

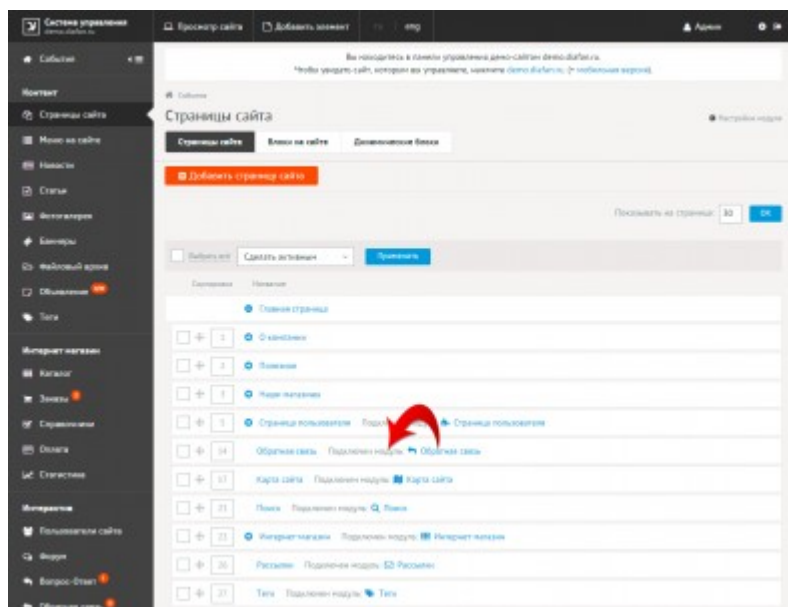
Проще всего разбираться с незнакомыми алгоритмами опираясь на готовые решения. Возьмем страницу списка новостей на стандартном дизайне DIAFAN.CMS. В демо-версии это страница категории [Новости туризма](#)

[Открыть пример страницы](#)

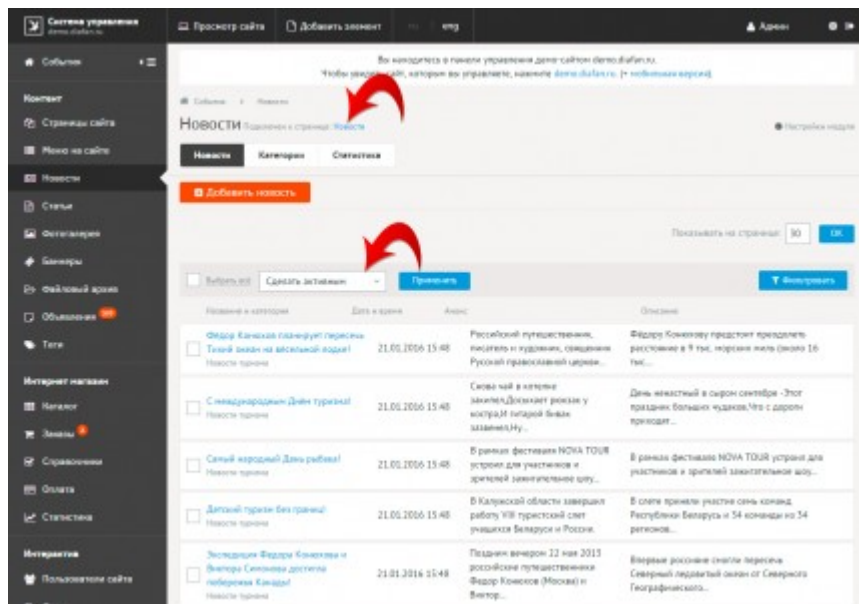
Чтобы понимать, каким образом DIAFAN.CMS строит эту страницу, нужно знать две вещи:

- Какой шаблон дизайна используется на текущей странице сайта
- Какой модуль подключен к текущей странице

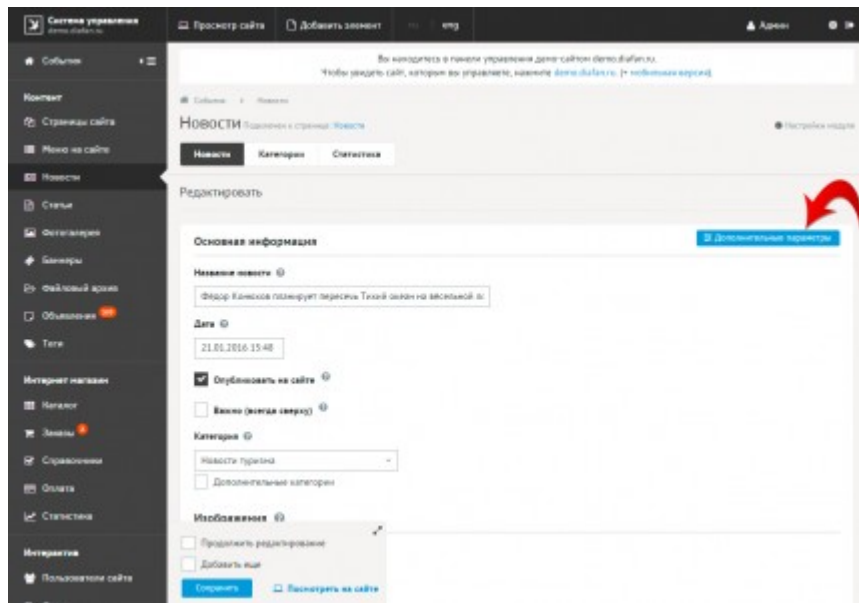
Поскольку наша страница находится *Главная / Полезное / Новости*, открываем **Страницы сайта** и находим её.



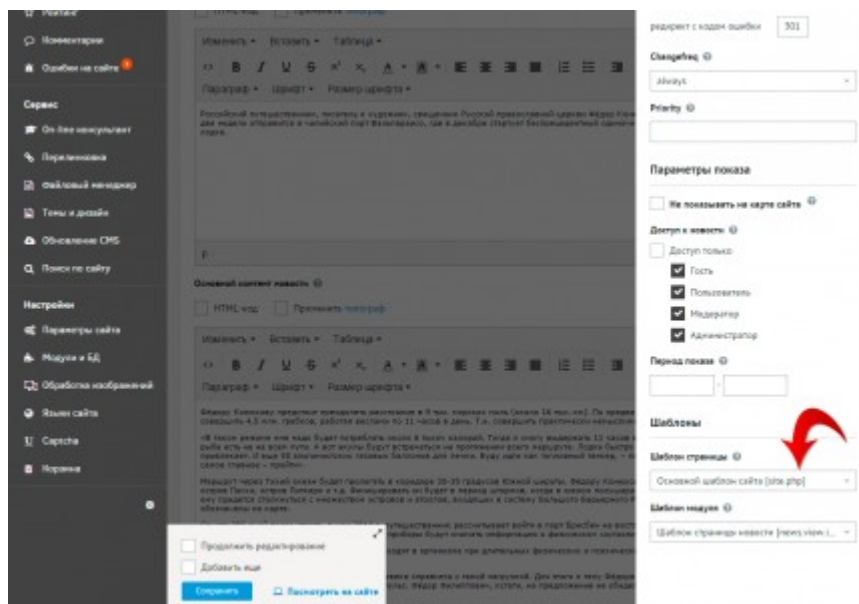
Если кликнуть на страницу с подключенным модулем, первым откроется список элементов модуля. Чтобы узнать, какой шаблон подключен к текущей странице сайта, нужно открыть её редактирование. Чтобы сделать это из списка элементов подключенного к странице модуля, нужно нажать *изменить*.



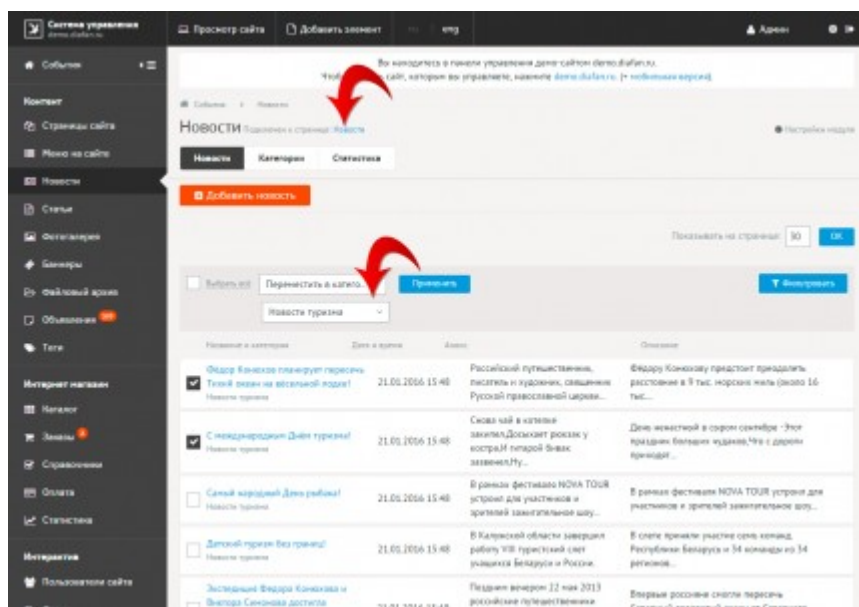
Назначение шаблона - не самая частоиспользуемая функция, поэтому она вынесена в раздел *Дополнительно*. Раскройте его.



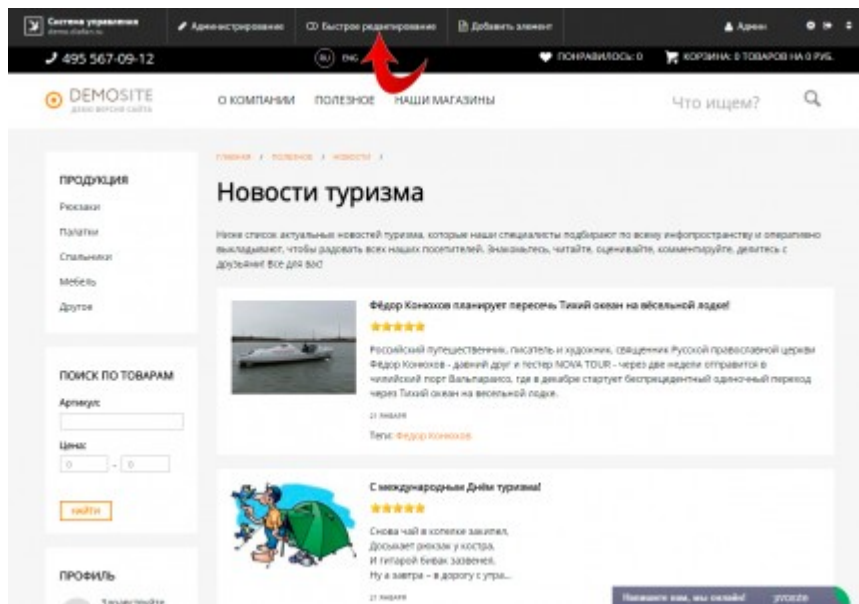
Вы увидите, какой шаблон назначен для страницы «Новости». В нашем случае это *themes/site.php*



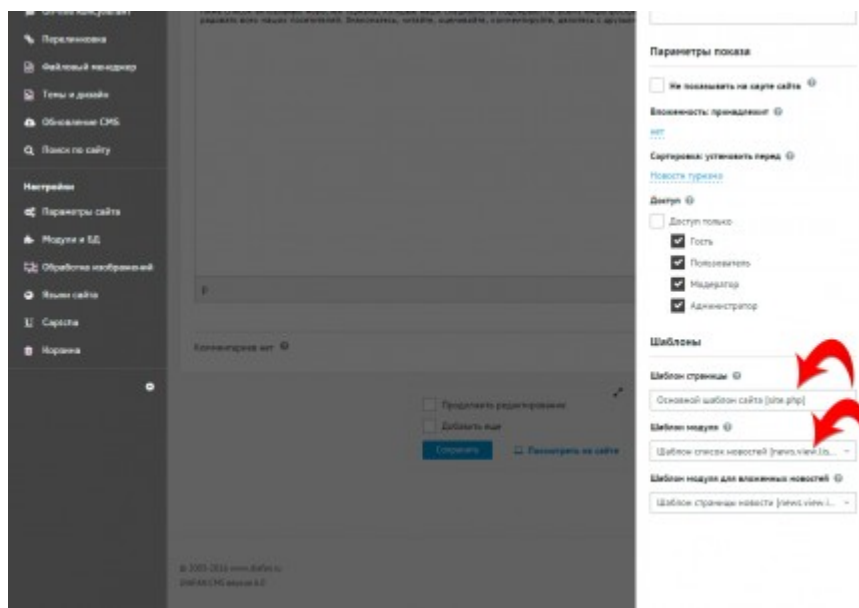
Далее. Поскольку мы рассматриваем новости из категории «Новости туризма», а не корневые для списка новостей, нам нужно знать, какие шаблоны назначены на эту категорию. Выбираем категорию «Новости туризма» над списком новостей и затем жмем *изменить*.



Если у Вас подключена [панель быстрого редактирования](#), найти текущую страницу гораздо проще. Просто кликните на иконку-карандашик на панели.



Категория «Новости туризма» откроется для редактирования. Раскрываем *Дополнительно* и видим, что для категории унаследовался основной шаблон сайта *themes/site.php*, а для модуля используется шаблон модуля новостей *modules/news/views/news.view.list.php*



Теперь можно смело открывать файл шаблона сайта *themes/site.php* и шаблон модуля новости и вносить там необходимые изменения. Для наглядности, мы изобразили связи между шаблонными тегами и блоками на сайте. А также указали, в каком файле правится каждый шаблонный тег.



[Открыть схему в исходном размере](#)

Внимание! В данном разделе документации говорится о стандартном дизайне и базовых шаблонах сайта. Это *themes/site.php*, *themes/site_start.php* и прочие, находящиеся в папке *themes*. А также шаблонах модулей, находящихся по адресам *modules/модуль/views/*. Однако, если на Вашем сайте используются темы, полученные в результате кастомизации, шаблоны сайта и шаблоны модулей могут находиться в папке **** custom/modules/модуль/views/ ****. [Подробнее о темах](#)

И еще раз приведем ссылку на [справочник всех шаблонных тегов](#)

Как подключить DIAFAN.CMS к дизайну сайта

- [Внедрение дизайна \(HTML-верстки\) в DIAFAN.CMS](#)
- [Мультидизайн в DIAFAN.CMS](#)

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

Внедрение дизайна (HTML-верстки) в DIAFAN.CMS

Все очень просто. Подключить дизайн к DIAFAN.CMS – легче легкого!

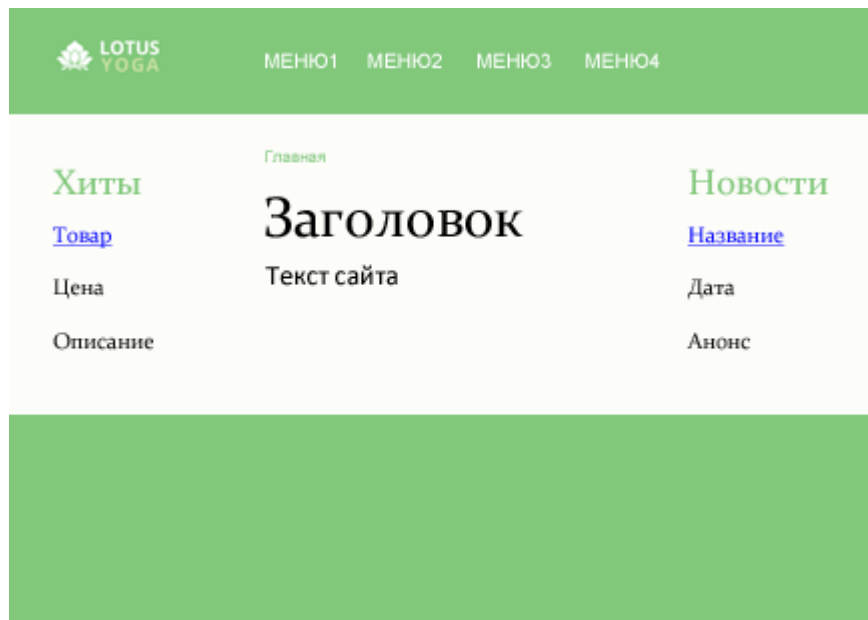
После [установки DIAFAN.CMS](#) на вашем сайте будет дизайн со стандартным дизайном системы, как в [демо-версии](#).

За этот дизайн отвечает базовый [шаблон](#), *themes/site.php*. По сути, это простой, обычный HTML-файл, но с [шаблонными тегами DIAFAN.CMS](#). Шаблонные теги «подтягивают» в шаблон сайта многочисленные шаблоны модулей. Смотрите подробнее, [как устроен стандартный шаблон](#).

Шаблонные теги DIAFAN.CMS имеют тот же синтаксис, что и обычные HTML-теги, поэтому корректно выглядят в визуальных редакторах веб-страниц, например, DreamViewer.

Совет: при внедрении своего дизайна лучше не удалять, а переименовать исходный шаблон *themes/site.php*, и затем оттуда копировать теги. А также добавьте в избранное [справочник шаблонных тегов](#).

Перейдем к делу. Начинать лучше от простого к сложному. Допустим, Ваш дизайн сайта выглядит так:



HTML-код этого дизайна после верстки примерно такой:

Пример:

```

<html>
<head>
  <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">
  <link href="css/style.css" rel="stylesheet" type="text/css">
</head>
<body>
  <div class="body">
    <div class="top">
      <div class="logo">
        <a href="#"></a>
      </div>
      <div class="menu">
        <a href="#">Меню 1</a>
        <a href="#">Меню 2</a>
        <a href="#">Меню 3</a>
        <a href="#">Меню 4</a>
      </div>
    </div>
    <div class="content">
      <div class="news">
        <h3>Хиты</h3>
        <a href="#">Товар</a>
        <p>Цена</p>
        <p>Описание</p>
      </div>
      <div class="center">
        <div class="nav">
          <a>Главная</a>
        </div>
        <div class="text">
          <h1>Заголовок</h1>
          <p>Текст сайта</p>
        </div>
      </div>
      <div class="right">
        <h3>Новости</h3>
        <a href="#">Название</a>
        <p>Дата</p>
        <p>Анонс</p>
      </div>
    </div>
  </div>
</body>
</html>

```

Первым делом переименуйте этот HTML файл и сохраните в DIAFAN.CMS как *themes/site.php*. Вы сразу увидите, что он отобразится на сайте, один в один. Ссылки выведутся как «#», тег

title как «Заголовок сайта» и т.д. Чтобы в этом дизайне DIAFAN.CMS выводила информацию из административной части, нужно расставить шаблонные теги так:

Пример:

```
<html>
<head>
    <!-- шаблонный тег генерирует все необходимые теги для шапки сайта, в том числе
    мета-теги. Код шаблонной функции в файле themes/functions/show_head.php -->
    <insert name="show_head">

    <!-- шаблонный тег подключает CSS-файлы из папки css, с учетом кастомизации,
    сжимает файлы. Код шаблонной функции в файле themes/functions/show_css.php -->
    <insert name="show_css" files="style.css">
</head>
<body>
    <div class="body">
        <div class="top">
            <div class="logo">
                <!-- шаблонный тег path выводит корректный путь до корня сайта (включая
                мобильную и мультиязычную версии сайта) -->
                <a href="<insert name="path">">img/logo.png"
                alt="Логотип"></a>
            </div>
            <div class="menu">
                <!-- шаблонный тег вывода пунктов меню. Код шаблонной функции в файле
                modules/menu/views/menu.view.show_block.php -->
                <insert name="show_block" module="menu" id="1">
            </div>
        </div>
        <div class="content">
            <div class="news">
                <!-- шаблонный тег вывода блока товаров, одного случайного. Код шаблонной
                функции в файле modules/shop/views/shop.view.show_block.php -->
                <insert name="show_block" module="shop" count="1" sort="rand">
            </div>
            <div class="center">
                <div class="nav">
                    <!-- шаблонный тег вывода навигации "Хлебные крошки". Код шаблонной
                    функции в файле themes/functions/show_breadcrumb.php -->
                    <insert name="show_breadcrumb">
                </div>
                <div class="text">
                    <!-- шаблонный тег вывода основного контента сайта. Код шаблонной функции
                    в файле themes/functions/show_body.php -->
                    <insert name="show_body">
                </div>
            </div>
            <div class="right">
                <!-- шаблонный тег вывода блока анонсов новостей, одной последней. Код
```

```
шаблонной функции в файле modules/news/views/news.view.show_block.php -->
    <insert name="show_block" module="news" count="1">
    </div>
</div>
</div>
<!-- шаблонный тег подключает JS-скрипты для корректной работы DIAFAN.CMS,
добавляется перед закрывающим </body>. Код шаблонной функции в файле
themes/functions/show_js.php -->
    <insert name="show_js">
</body>
</html>
```

Всё.

Чтобы затем провести полировку дизайна, например, подправить блок вывода новостей, выводимых тегом `<insert name="show_block" module="news" count="1">` в шаблоне модуля / *modules/news/view/news.view.show_block.php*

Надеемся, что принцип понятен. Далее можно только расширять количество используемых шаблонных тегов и параметров в них. Выводить 3 новости, а не одну, выводить блок новостей с картинками или без, выводить случайные товары или лидеры продаж и т.д.

Внимание! Важно знать, что после получения [обновлений DIAFAN.CMS](#) или [генерации темы](#), адреса файлов Вашего внедренного дизайна **могут измениться!**

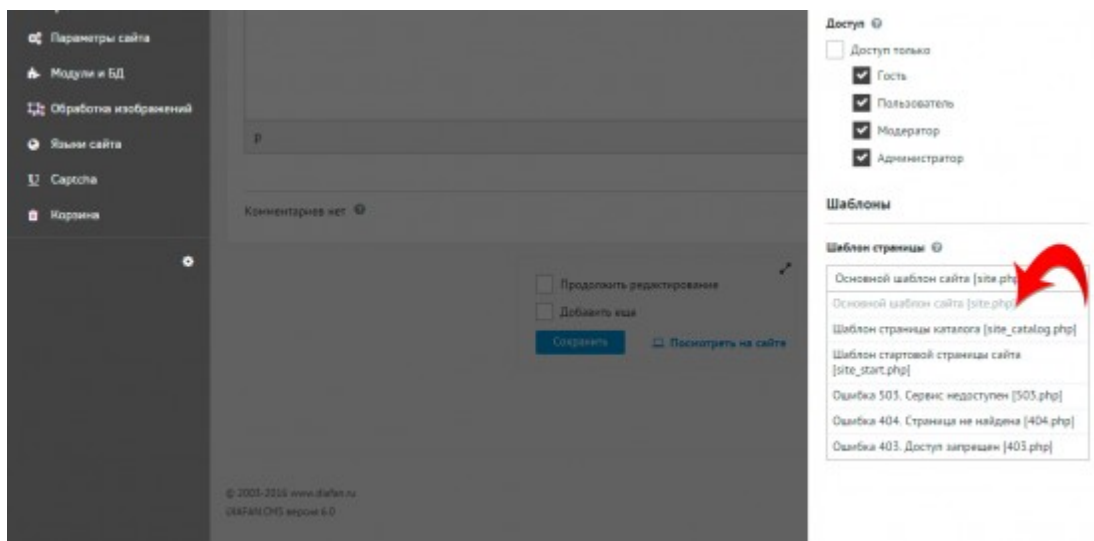
Полное описание всех шаблонных тегов и их параметров смотрите [здесь](#).

Мультидизайн в DIAFAN.CMS

Можно заметить, что в папке *themes* не один шаблон сайта, а несколько. Есть *themes/site.php*, *themes/site_start.php* или *themes/site_catalog.php*.

Несмотря на то, что большинство задач по внедрению различий в дизайне на разных страницах сайта можно решить с помощью [динамических блоков](#), используя всего один шаблон, можно использовать несколько шаблонов и назначать их разным страницам по своему усмотрению.

Если для какой-то отдельной страницы на сайте необходимо иметь совершенно иной от общего дизайн, сохраните шаблон в папке *themes* (или в папке активной [темы](#)) с расширением *php*. Затем откройте нужную страницу на редактирование в административной части сайта, и раскрыв вкладку «Дополнительные параметры», укажите загруженный шаблон в выпадающем списке «Шаблон дизайна».



Этот шаблон будет применяться ко всем вновьсоздаваемым дочерним страницам.

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.

Страницы админки

Редактирование страниц в административной части, к которым подключены модули. Формирование на их основе меню в административной части.

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, кастомизированные темы.

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

- [Подключение](#)
- [Административная часть](#)
- [Блоки на сайте](#)
- [Динамические блоки](#)
- [Настройки](#)
- [Шаблонные теги](#)
- [База данных](#)
- [Файлы](#)

Модуль «Страницы сайта» – основной механизм управления сайтом.

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

Подключение

Подключение страниц сайта используется в первую очередь для получения данных о текущей странице.

Подключаемая часть – файл *modules/site/site.inc.php*. В нем описан класс *Site_inc*. В модуле к объекту класса можно обратиться через переменную `$this->diafan->_site`. Экземпляр класса создается при первом вызове переменной.

Свойства

var **id** – номер текущей страницы, уникальный идентификатор каждой страницы сайта.

Пример:

```
$this->diafan->_site->id;
```

var **name** – название текущей страницы.

Пример:

```
$this->diafan->_site->name;
```

var **keywords** – данные из meta-тега *keywords* текущей страницы.

Пример:

```
$this->diafan->_site->keywords;
```

var **descr** – данные из meta-тега *description* текущей страницы.

Пример:

```
$this->diafan->_site->descr;
```

var **parent_id** – номер страницы родителя.

Пример:

```
$this->diafan->_site->parent_id;
```

var **title_no_show** – не показывать заголовков h1 текущей страницы, да/нет (1/0).

Пример:

```
$this->diafan->_site->title_no_show;
```

var **noindex** – не индексировать текущую страницу, да/нет (1/0).

Пример:

```
$this->diafan->_site->noindex;
```

var **title_meta** – заголовок текущей страницы из тега *title*.

Пример:

```
$this->diafan->_site->title_meta;
```

var **canonical** – канонический тег для текущей страницы.

Пример:

```
$this->diafan->_site->canonical;
```

var **timeedit** – время редактирования текущей страницы, в UNIX-формате.

Пример:

```
$this->diafan->_site->timeedit;
```

var **theme** – имя файла шаблона дизайна текущей страницы.

Пример:

```
$this->diafan->_site->theme;
```

var **js** – JavaScript-код.

Пример:

```
$this->diafan->_site->js;
```

var **module** – модуль, прикрепленный к текущей странице.

Пример:


```
$this->diafan->_site->module;
```

var **text** – контент текущей страницы.

Пример:

```
$this->diafan->_site->text;
```

var **rewrite** – ЧПУ текущей страницы, для страницы *http://site.ru/news/popular/novost/* в переменной будет "news/popular/novost".

Пример:

```
$this->diafan->_site->rewrite;
```

Вышеописанные свойства могут быть заданы при определении текущей страницы сайта. Затем в прикрепленном модуле могут быть определены следующие свойства.

var **breadcrumb** – часть навигации «Хлебные крошки».

Пример:

```
$this->diafan->_site->breadcrumb;
```

var **titlemodule** – заголовок страницы, сформированный автоматически прикрепленным модулем.

Пример:

```
$this->diafan->_site->titlemodule;
```

var **titlemodule_meta** – заголовок текущей страницы для тега *title*, сформированный прикрепленным модулем.

Пример:

```
$this->diafan->_site->titlemodule_meta;
```

var **hide_previous_next** – спрятать ссылки на предыдущую, последующую страницы, да/нет (1/0).

Пример:

```
$this->diafan->_site->hide_previous_next;
```

var **css_view = array()** – CSS-файлы, подключаемые в модулях.

var **js_view = array()** – JS-скрипты, подключаемые в модулях.

Пример:

```
// добавляем JS-файл для подключения в конце страницы  
$this->diafan->_site->js_view[] = 'modules/search/js/search.show_search.js';
```

Кроме этих свойств с помощью служебной функции `__get()` можно задать любое свойство страницы и затем его использовать.

var **js_code** = **array()** – JS-код, определяемый в модулях.

var **nocache** = **false** – страница не кэшируется при включенном экстремальном кэшировании.

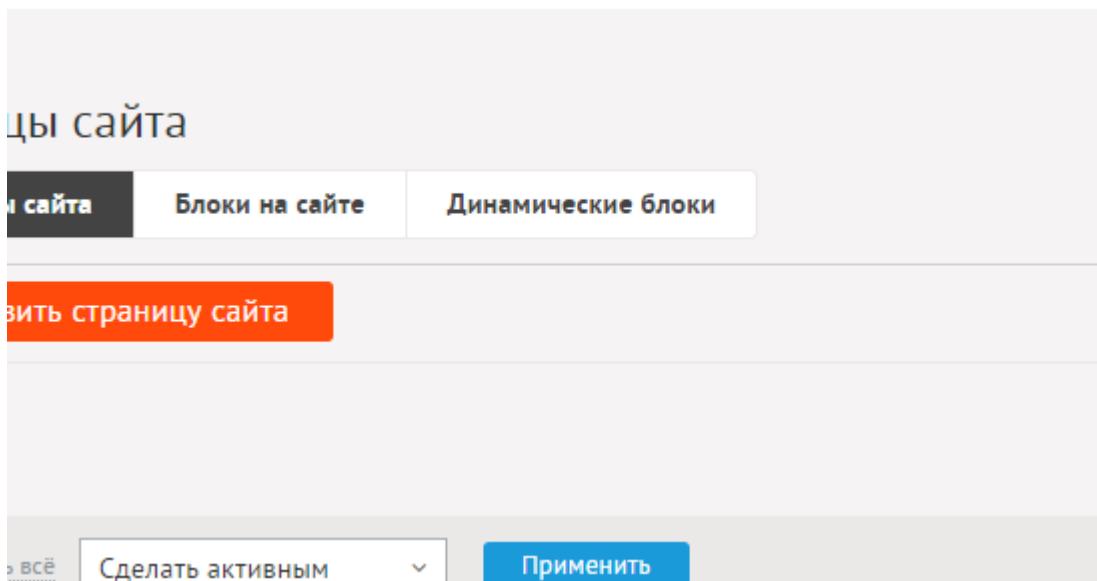
Методы

void **set ()** – Определяет страницу сайта, задает параметры страницы.

Пример:

```
// в файле includes/init.php  
$this->_site->set();
```

Административная часть



Если к странице подключен какой-либо модуль, то возле страницы выведется иконка модуля, а нажатие на название страницы переводит сразу на управление этим модулем. Саму страницу, к которой подключен модуль, тогда можно найти по ссылке «изменить» рядом с заголовком модуля.

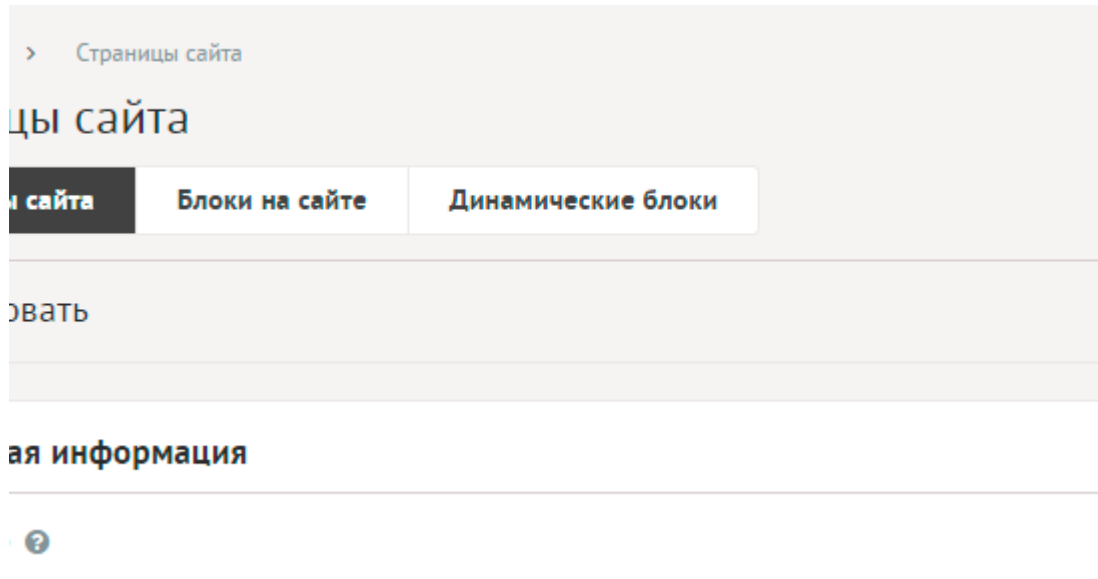
Пример:

Если в структуре сайта есть страница «Наши новости», то нажатие на её название приведет к редактированию этой страницы. А если к странице «Наши новости» прикрепить модуль «Новости», то нажатие приведет к списку новостей из модуля

«Новости». Отредактировать же страницу «Наши новости» можно нажав на «(изменить)» рядом с заголовком перед списком новостей.

Для редактирования страницы необходимо нажать на нее.

Дополнительные параметры разворачиваются отдельно, располагаются под основной структурой.



Страница сайта имеет следующие характеристики.

Основные:

- **Название** – Название страницы, отображается в ссылках на страницу, используется для автоматической генерации пункта меню, заголовка страницы и её адреса (ЧПУ). Параметр имеет разные значения для разных языковых версий сайта.
- **Опубликовать на сайте** – Отображение страницы на сайте. Если не отмечена, страница на сайте не будет доступна пользователям и выдаст «404 Страница не найдена». Страница всегда доступна для просмотра администратору. 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 страницу.
- **Корневая страница** – Перемещение текущей страницы и всех её подстраниц в принадлежность другой страницы (администратору 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».
- **Контент страницы** – Основное содержимое страницы. Если отметить «Применить типограф», контент будет отформатирован согласно правилам экранной типографики с помощью [веб-сервиса «Типограф»](#). Опция «HTML-код» позволяет отключить визуальный редактор для текущего поля. Значение этой настройки будет учитываться и при последующем редактировании. Параметр имеет разные значения

для разных языковых версий сайта.

- **Динамические блоки.**
- **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.

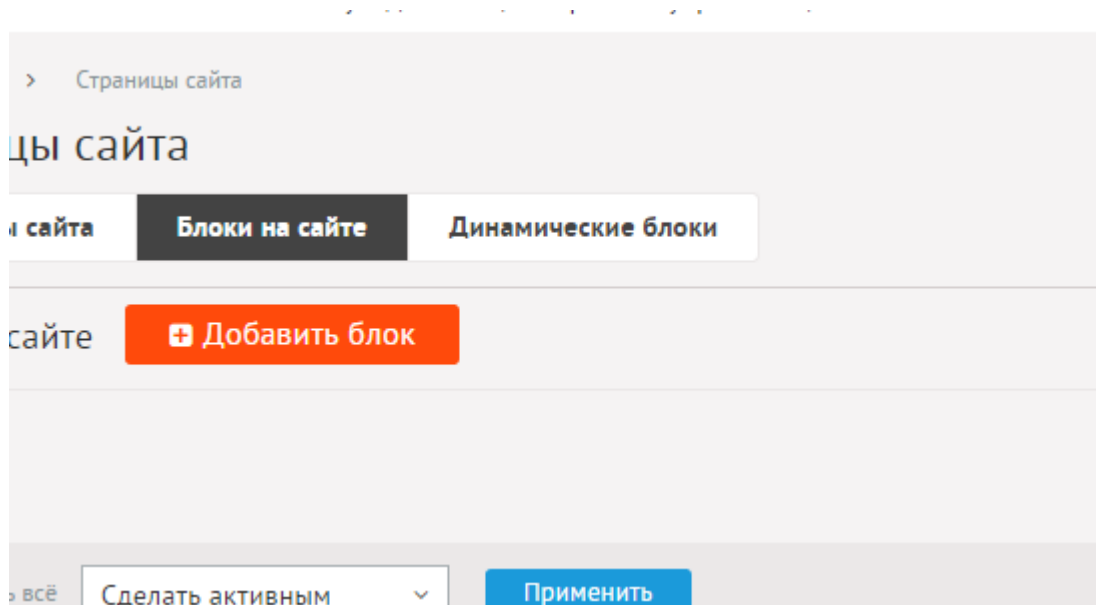
- **Теги** – Добавление тегов к странице. Можно добавить либо новый тег, либо открыть и выбрать из уже существующих тегов. Параметр выводится, если в настройках модуля включен параметр «Подключить теги».
- **Комментарии** – Комментарии, которые оставили пользователи к текущей странице. Параметр выводится, если в настройках модуля включен параметр «Показывать комментарии к новостям».

Дополнительные:

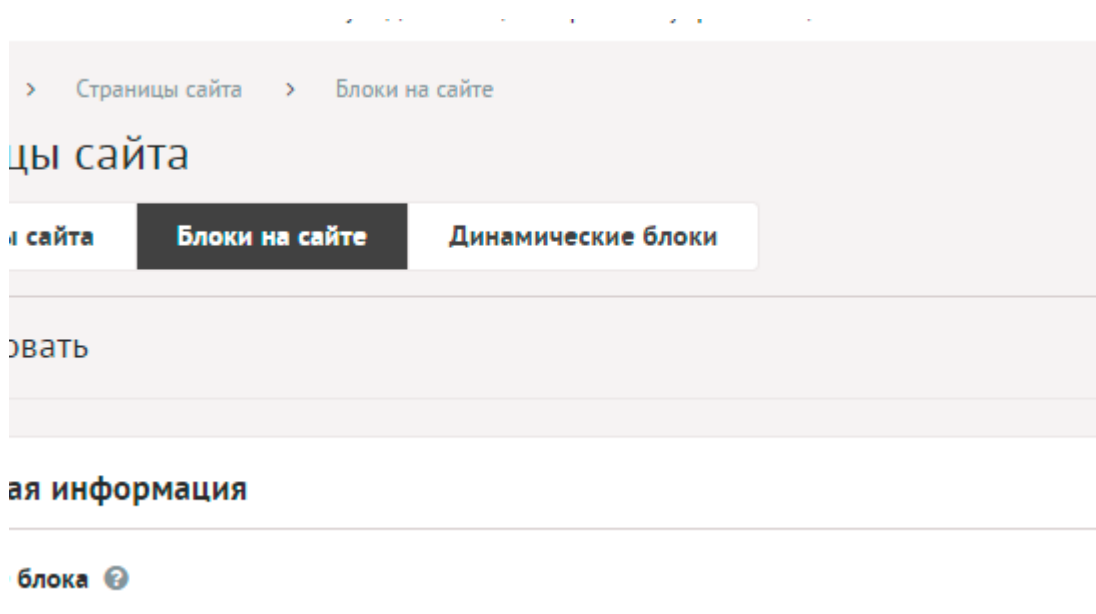
- **Номер** – Номер страницы в БД (веб-мастеру и программисту).
- **Редактор** – Изменяется после первого сохранения. Показывает, кто из администраторов сайта первый правил текущую страницу.
- **Время последнего изменения** – Изменяется после сохранения элемента. Отдается в заголовке *Last Modify*.
- **Параметры SEO.**
- **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, тэг 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.
- **Канонический тег** – URL канонической страницы вида: <http://site.ru/pseudossylka/>, на которую переносится "ссылочный вес" данной страницы. Используется для страниц с похожим или дублирующимся контентом (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).
- **Параметры показа.**
- **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.
- **Шаблоны.**
- **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!).
- **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*.

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.
- **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»](#). Опция «HTML-код» позволяет отключить

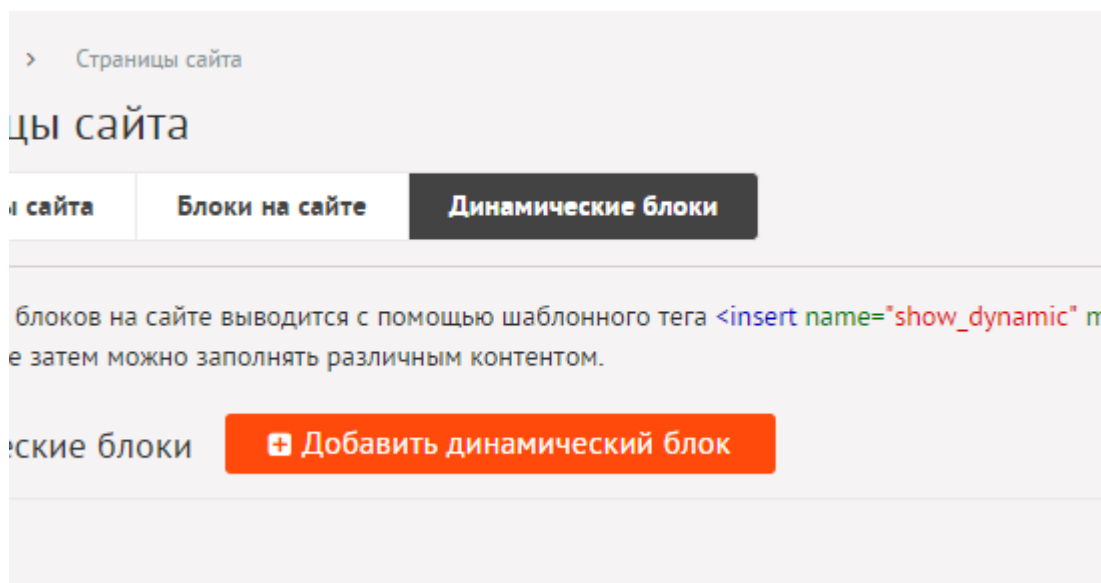
визуальный редактор для текущего поля. Значение этой настройки будет учитываться и при последующем редактировании. Параметр имеет разные значения для разных языковых версий сайта.

- **Отображать на страницах** – Выбор отдельных страниц, где будет показываться блок. Удерживайте CTRL, чтобы выбрать несколько страниц.

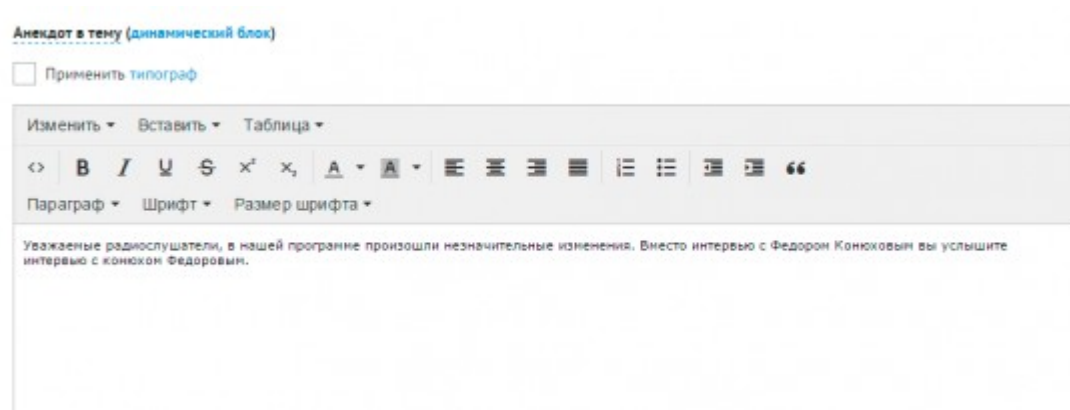
Дополнительные:

- **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).

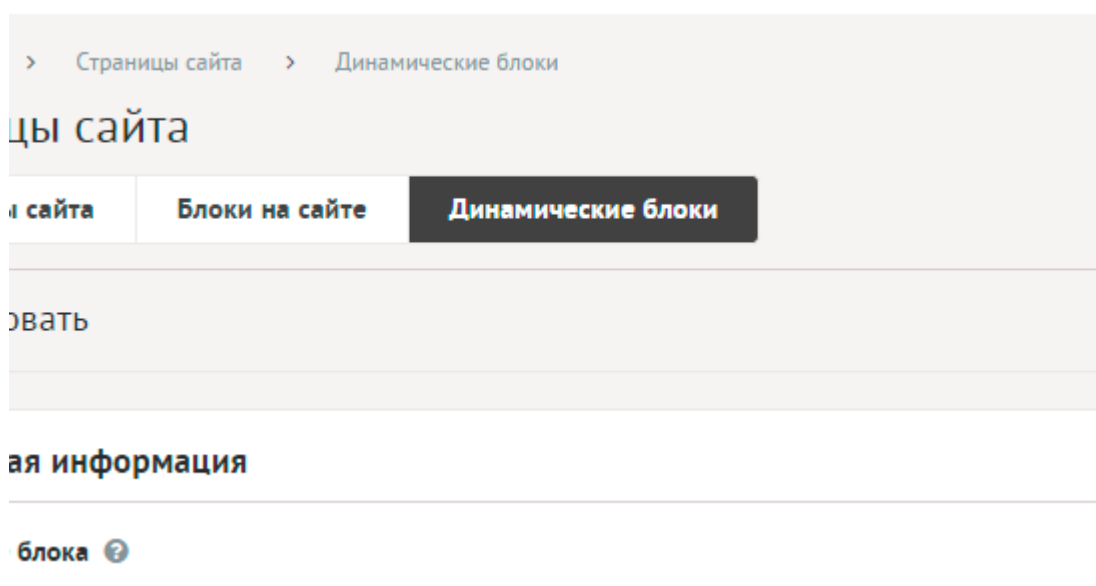
Динамические блоки



Динамические блоки – это контент, который может быть выведен шаблонным тегом [show_dynamic](#) в любом месте шаблона. В отличие от обычных блоков контент динамического блока заполняется непосредственно в элементах модуля (странице сайта, новости, категории новостей и т. д.) и уникален для каждой страницы. Содержание динамического блока можно применить на детей или вложенные элементы (например, для категорий товаров на вложенные категории или принадлежащие категории товары).



При редактировании динамического блока заполняются следующие его свойства.



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.
- **Не выводить название блока** – Если отмечено, заголовок перед содержимым блока автоматически выводиться не будет.
- **Тип** – Тип динамического блока. Возможные значения:
 - строка,
 - число,
 - дата,
 - дата и время,
 - текстовое поле,
 - поле с визуальным редактором,
 - электронный ящик.
- **Прикрепить к модулям** – Редактировать поле только в указанных модуля и для

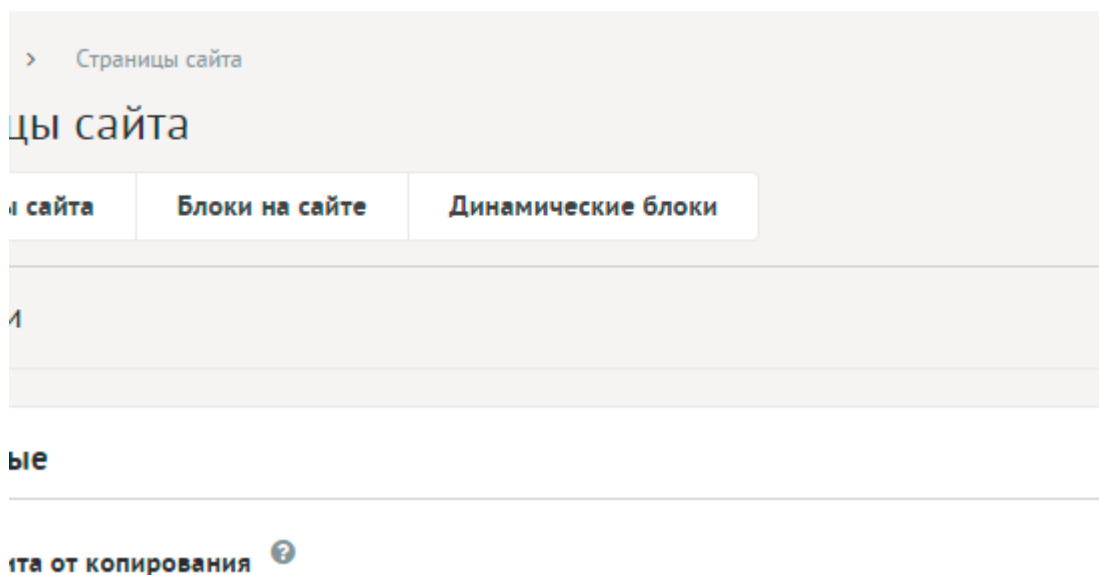
указанных типов элементов модуля. Возможные значения:

- элементам,
- категориям,
- брендам.
- **Подсказка для поля** – Будет выведено при редактировании содержимого блока в модуле.

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



- **Основные.**
- **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»](#).
- **Подключить перелинковку** – Отображение перелинковки в модуле. Подробности см. в разделе [модуль «Перелинковка»](#).
- **Изображения.**
- **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».
- **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».
- **Автогенерация для SEO.**
- **Шаблон для автоматического генерирования Title** – Если шаблон задан и для страницы не прописан заголовок *Title*, то заголовок автоматически генерируется по шаблону. В шаблон можно добавить:
 - %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 и для страницы не заполнено поле *Keywords*, то поле *Keywords* автоматически генерируется по шаблону. В шаблон можно добавить:
 - %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 и для страницы не заполнено поле *Description*, то поле *Description* автоматически генерируется по шаблону. В шаблон можно добавить:
 - %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 блока на сайте, номер которой передан в виде атрибута `id`.

Атрибуты:

- **id** – идентификатор блока;
- **defer** – маркер отложенной загрузки шаблонного тега: **event** – загрузка контента только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** – загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка контента только по желанию пользователя;
- **defer_title** – текстовая строка, выводимая на месте появления загружаемого контента с помощью отложенной загрузки шаблонного тега;
- **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»

В шаблоне тега можно получить значение любого атрибута через переменную `$result["attributes"]["название атрибута"]`.

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

Атрибуты:

- **defer** – маркер отложенной загрузки шаблонного тега: **event** – загрузка контента только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** – загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка контента только по желанию пользователя;
- **defer_title** – текстовая строка, выводимая на месте появления загружаемого контента с помощью отложенной загрузки шаблонного тега;

— **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

В шаблоне тега можно получить значение любого атрибута через переменную `$result["attributes"]["название атрибута"]`.

show_dynamic – выводит содержимое динамического блока, номер которой передан в виде атрибута `id`.

Атрибуты:

- **id** – идентификатор динамического блока;
- **element_id** – номер элемента, для которого будет выведено значение блока, по умолчанию текущий элемент;
- **module_name** – модуль элемента, для которого будет выведено значение блока, по умолчанию текущий модуль;
- **element_type** – тип элемента, для которого будет выведено значение блока, по умолчанию тип текущего элемента;
- **defer** – маркер отложенной загрузки шаблонного тега: **event** – загрузка контента только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** – загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка контента только по желанию пользователя;
- **defer_title** – текстовая строка, выводимая на месте появления загружаемого контента с помощью отложенной загрузки шаблонного тега;
- **template** – шаблон тега (файл `modules/site/views/site.view.show_dynamic_`**template**`.php`; по умолчанию шаблон `modules/site/views/site.view.show_dynamic.php`).

Пример:

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

В шаблоне тега можно получить значение любого атрибута через переменную `$result["attributes"]["название атрибута"]`.

show_images – выводит изображения, прикрепленные к странице сайта, если в

конфигурации модуля «Страницы сайта» включены параметры «Использовать изображения».

Атрибуты:

- **defer** – маркер отложенной загрузки шаблонного тега: **event** – загрузка контента только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** – загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка контента только по желанию пользователя;
- **defer_title** – текстовая строка, выводимая на месте появления загружаемого контента с помощью отложенной загрузки шаблонного тега;
- **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 группу фотографий, прикрепленных к текущей странице

В шаблоне тега можно получить значение любого атрибута через переменную `$result["attributes"]["название атрибута"]`.

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

Атрибуты:

- **defer** – маркер отложенной загрузки шаблонного тега: **event** – загрузка контента только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** – загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка контента только по желанию пользователя;
- **defer_title** – текстовая строка, выводимая на месте появления загружаемого контента с помощью отложенной загрузки шаблонного тега;
- **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">
```

В шаблоне тега можно получить значение любого атрибута через переменную `$result["attributes"]["название атрибута"]`.

show_previous_next – выводит ссылки на предыдущую и последующую страницы.

Атрибуты:

- **defer** – маркер отложенной загрузки шаблонного тега: **event** – загрузка контента только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** – загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка контента только по желанию пользователя;
- **defer_title** – текстовая строка, выводимая на месте появления загружаемого контента с помощью отложенной загрузки шаблонного тега;
- **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">
```

В шаблоне тега можно получить значение любого атрибута через переменную `$result["attributes"]["название атрибута"]`.

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

Атрибуты:

- **defer** – маркер отложенной загрузки шаблонного тега: **event** – загрузка контента только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** – загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка контента только по желанию пользователя;
- **defer_title** – текстовая строка, выводимая на месте появления загружаемого контента с помощью отложенной загрузки шаблонного тега;
- **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

В шаблоне тега можно получить значение любого атрибута через переменную `$result["attributes"]["название атрибута"]`.

База данных

[{site}](#) – Страницы сайта

[{site_blocks}](#) – Блоки на сайте

[{site_blocks_site_rel}](#) – Данные о том, на каких страницах выводятся блоки на сайте

[{site_dynamic}](#) – Динамические блоки

[{site_dynamic_element}](#) – Контент динамических блоков, заполненный в элементах модулей

[{site_dynamic_module}](#) – Данные о том, в каких модулях заполняются динамические блоки

[{site_parents}](#) – Родительские связи страниц сайта

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/admin/site.admin.dynamic.php* – редактирование динамических блоков на сайте;
5. *modules/site/admin/site.admin.inc.php* – подключение модуля к административной части других модулей;
6. *modules/site/site.php* – контроллер;
7. *modules/site/site.inc.php* – подключение модуля;
8. *modules/site/site.install.php* – установка модуля;
9. *modules/site/site.model.php* – модель;
10. *modules/site/site.search.php* – настройки для поисковой индексации для модуля «Поиск»;
11. *modules/site/site.sitemap.php* – карта ссылок для модуля «Карта сайта»;
12. *modules/site/views/m/site.view.show_links.php* – шаблон вложенных страниц сайта;

13. *modules/site/views/site.view.show_block.php* – шаблон блока на сайте;
14. *modules/site/views/site.view.show_comments.php* – шаблон комментариев к странице сайта;
15. *modules/site/views/site.view.show_dynamic.php* – шаблон динамического блока;
16. *modules/site/views/site.view.show_images.php* – шаблон изображений к странице сайта;
17. *modules/site/views/site.view.show_links.php* – шаблон вложенных страниц сайта;
18. *modules/site/views/site.view.show_previous_next.php* – шаблон ссылок на предыдущую и следующую страницы сайта;
19. *modules/site/views/site.view.show_tags.php* – шаблон тегов к странице сайта.

Страницы админки

- [Подключение](#)
- [Административная часть](#)
- [База данных](#)
- [Файлы](#)

Модуль «Страницы админки» – структура страниц административной части, на их основе формируется меню административной части. Модуль является обязательным для установки.

По умолчанию модуль скрыт в системе, можно его открыть по ссылке:
site.ru/папка_административной_части/admin/

Подключение

Подключение используется в первую очередь для получения данных о текущей странице.

Подключаемая часть – файл *modules/admin/admin.inc.php*. В нем описан класс *Admin_inc*. В модуле к объекту класса можно обратиться через переменную `$this->diafan->_admin`. Экземпляр класса создается при первом вызове переменной.

Свойства

var **id** – номер текущей страницы, уникальный идентификатор каждой страницы.

Пример:

```
$this->diafan->_admin->id;
```

var **name** – название текущей страницы.

Пример:

```
$this->diafan->_admin->name;
```

var **title_module** – название текущего модуля.

var **docs** – ссылка на документацию для текущей страницы.

Пример:

```
$this->diafan->_admin->docs;
```

var **parent_id** – номер страницы родителя.

Пример:

```
$this->diafan->_admin->parent_id;
```

var **module** – модуль, прикрепленный к текущей странице.

Пример:

```
$this->diafan->_admin->module;
```

var **rewrite** – ЧПУ текущей страницы, для страницы *http://site.ru/admin/news/category/* в переменной будет "news/category".

Пример:

```
$this->diafan->_admin->rewrite;
```

Вышеописанные свойства могут быть заданы при определении текущей страницы. Затем в прикрепленном модуле могут быть определены следующие свойства.

var **css_view = array()** – CSS-файлы, подключаемые в модулях.

var **js_view = array()** – JS-скрипты, подключаемые в модулях.

Пример:

```
// добавляем JS-файл для подключения в конце страницы  
$this->diafan->_admin->js_view[] =  
'modules/attachments/admin/js/attachments.admin.inc.js';
```

Кроме этих свойств с помощью служебной функции `__get()` можно задать любое свойство страницы и затем его использовать.

var **js_code = array()** – JS-код, определяемый в модулях.

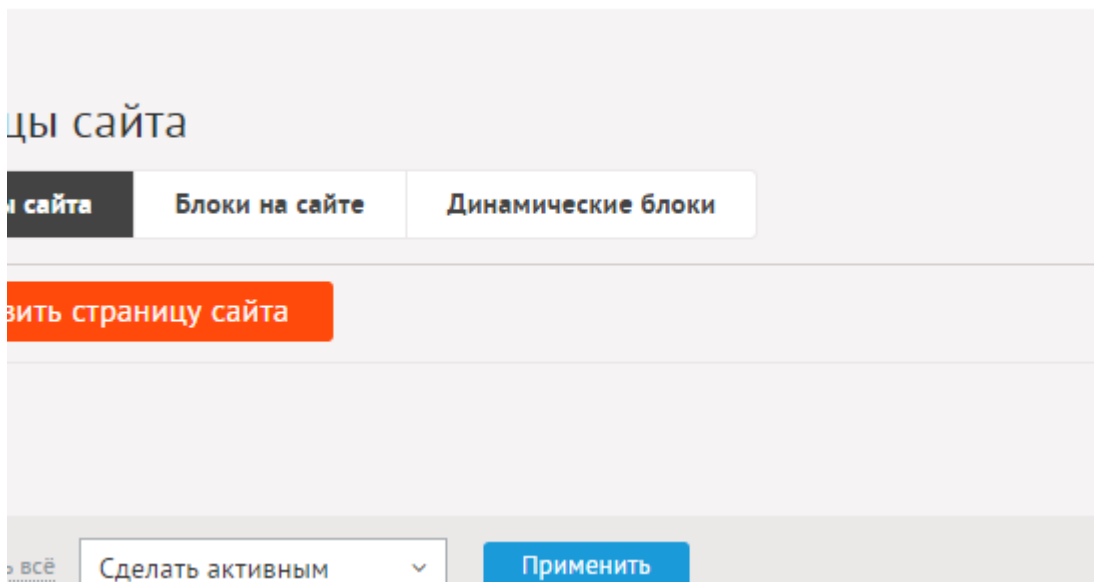
Методы

void **set ()** – Определяет страницу административной части, задает параметры страницы.

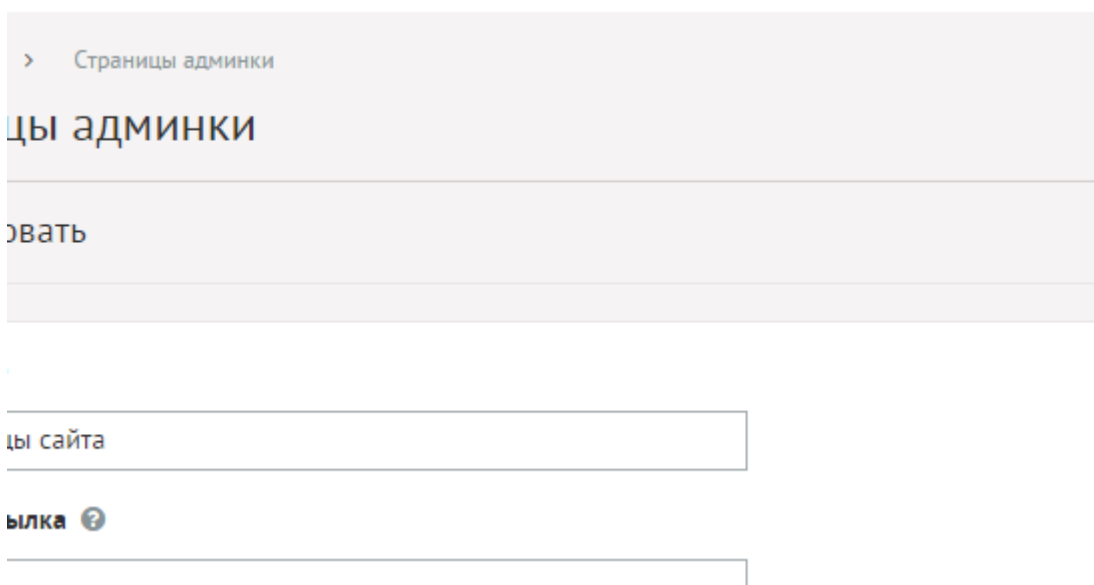
Пример:

```
// в файле adm/includes/init.php  
$this->_admin->set();
```

Административная часть



Страница административной части имеют следующие характеристики.



Основные:

- **Название.**
- **Псевдоссылка** – ЧПУ, адрес страницы вида: *http://site.ru/admin/pseudossylka/*.
- **Группа** – Логический блок в меню административной части. Возможные значения:
 - Контент,
 - Интернет магазин,
 - Интерактив,
 - Сервис,
 - Настройки.
- **Показывать в меню** – Возможность показать/скрыть в меню административной части.
- **Ссылка на документацию** – Ссылка выводится в подвале сайта.
- **Вложенность: принадлежит** – Перемещение текущей страницы в принадлежность

другой страницы.

- **Сортировка: установить перед** – Изменить положение текущей страницы среди других страниц в меню.
- **Ссылка на добавление элемента в быстром меню.**
- **Текст ссылки на добавление элемента в быстром меню** – Параметр появляется, если отмечена опция «Ссылка на добавление элемента в быстром меню».

База данных

[{admin}](#) – Страницы в админки

[{admin_parents}](#) – Родительские связи страниц админки

Файлы

1. *modules/admin/admin.inc.php* – подключение модуля;
2. *modules/admin/admin.install.php* – установка модуля;
3. *modules/admin/admin/admin.admin.php* – редактирование страниц административной части сайта;
4. *modules/admin/admin/js/admin.admin.js* – редактирование страниц административной части сайта, JS-сценарий.

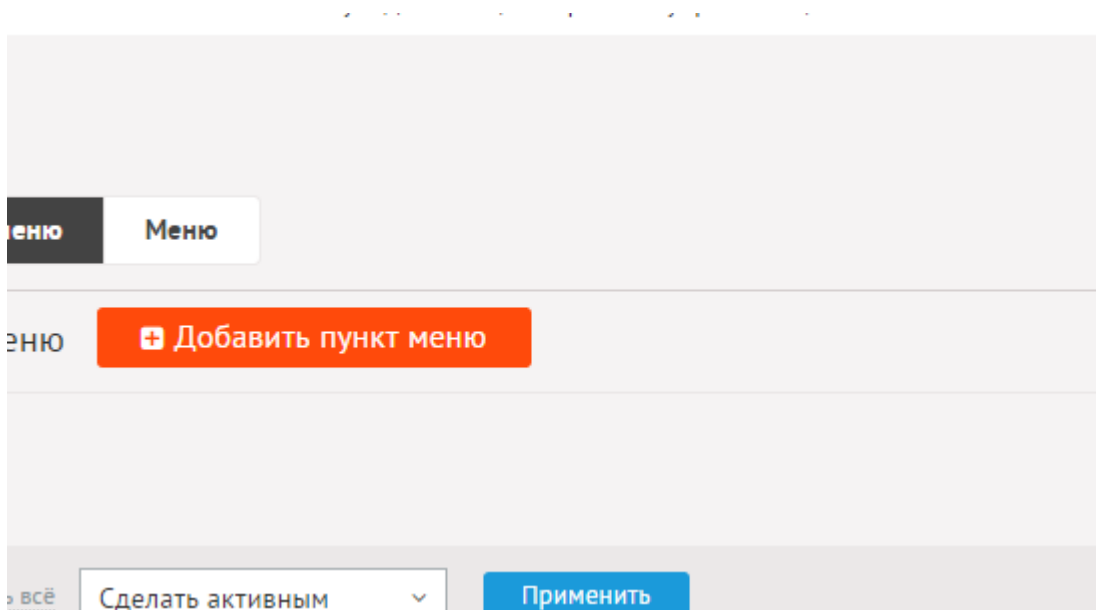
Меню на сайте

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

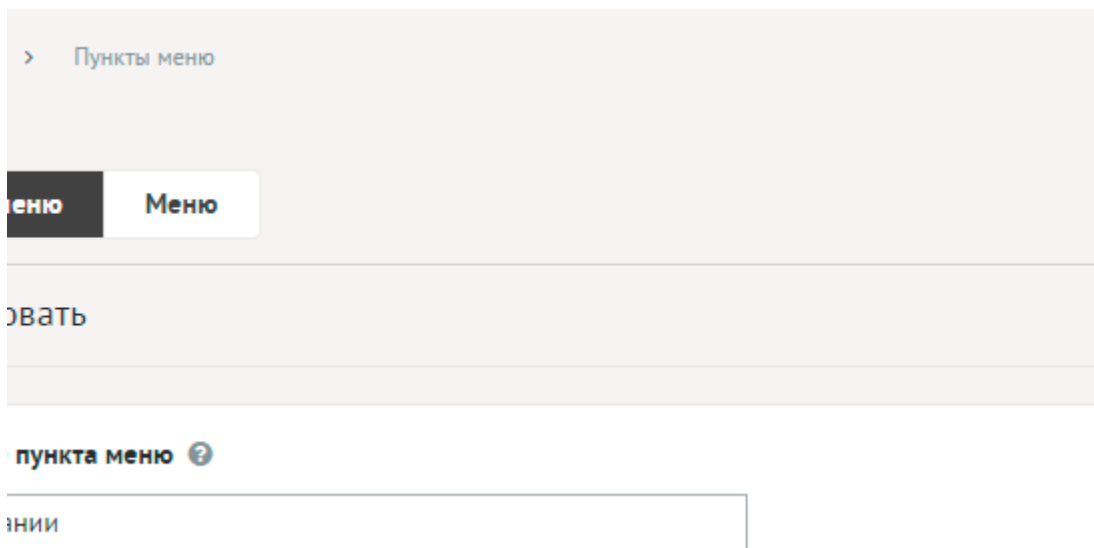
Модуль «Меню на сайте» формирует меню любой сложности и в любом количестве.

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.



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 с пункта меню** – Адрес ссылки, куда ведет текущий пункт меню. Заполняется автоматически при отметке галки «Показывать в меню» у элемента на сайте. Возможно указать ручную, как на внутреннюю страницу сайта, так и на любой другой сайт.
- **Открывать в новом окне** – Если отмечена, клик пользователя по пункту меню на сайте откроет ссылку в новом окне.
- **Атрибуты HTML ссылки** – HTML-код, выводимый в виде атрибутов для тега внутри ссылки `<a *** >` (Веб-мастеру и программисту).
- **Свойства пункта.**
- **Принадлежит к меню** – Выбор меню, к которому относится текущий пункт.
- **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. Тогда ссылкой будет не название пункта, а прикрепленное изображение.
- **Описание** – Краткое описание выводится для пункта меню на сайте. Параметр имеет разные значения для разных языковых версий сайта.

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.

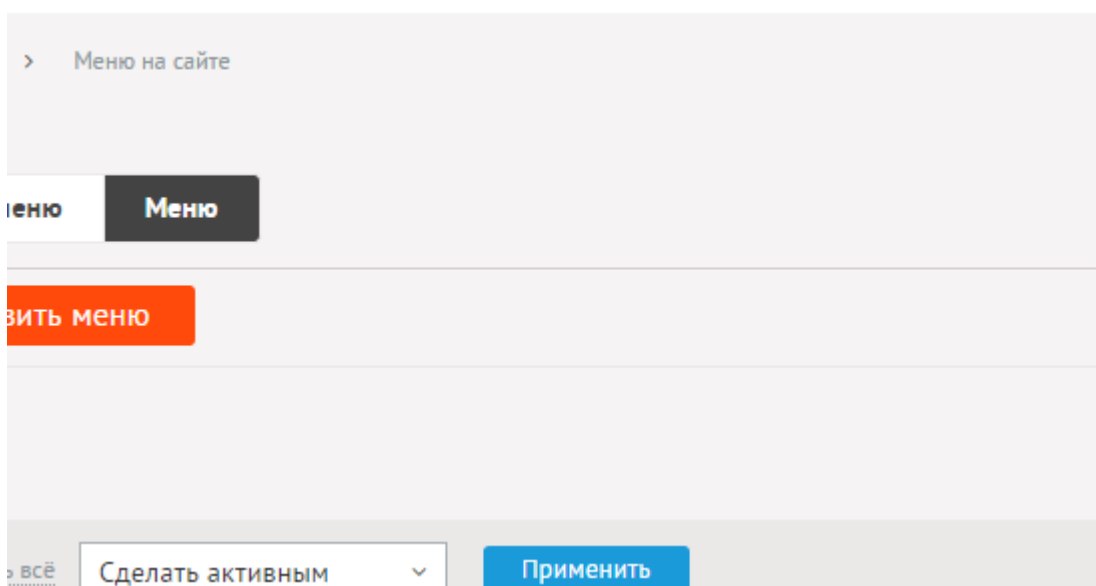
Структуру меню можно формировать непосредственно из модуля «Меню на сайте» или при заполнении страниц сайта, категорий или элементов модулей, отмечая галку «Показывать в меню». При этом структура сайта повторяется в меню.

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.

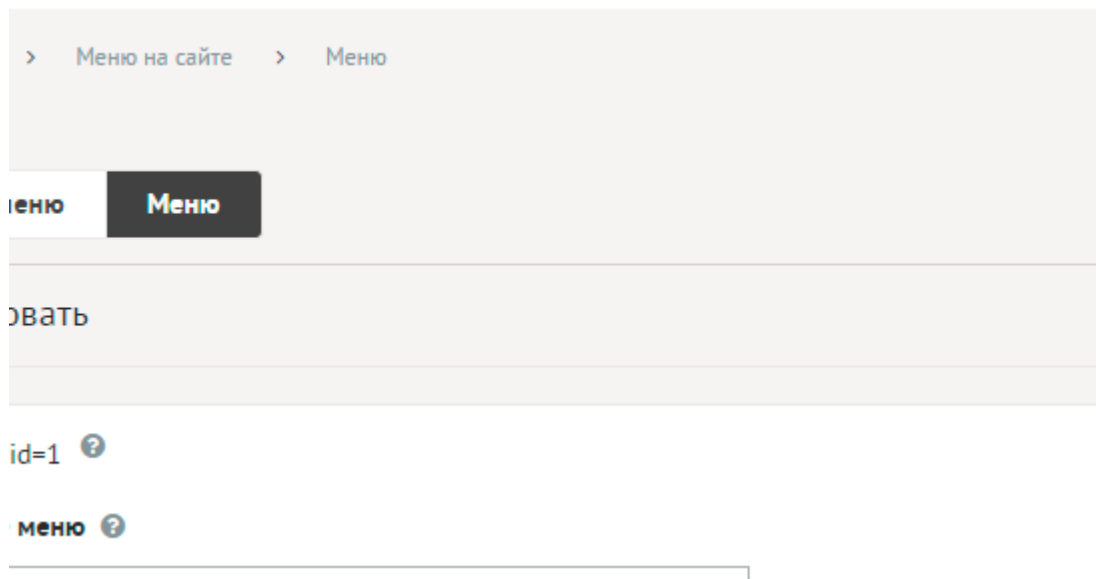
При формировании списка элементов, которые можно выбрать в качестве ссылок для пунктов меню, используются элементы других модулей. Ссылки на эти элементы формируются с помощью файла *modules/модуль/admin/модуль.admin.menu.php*.

Меню



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

При удалении меню удаляются все принадлежащие ему пункты меню.



Категории меню имеют следующие характеристики:

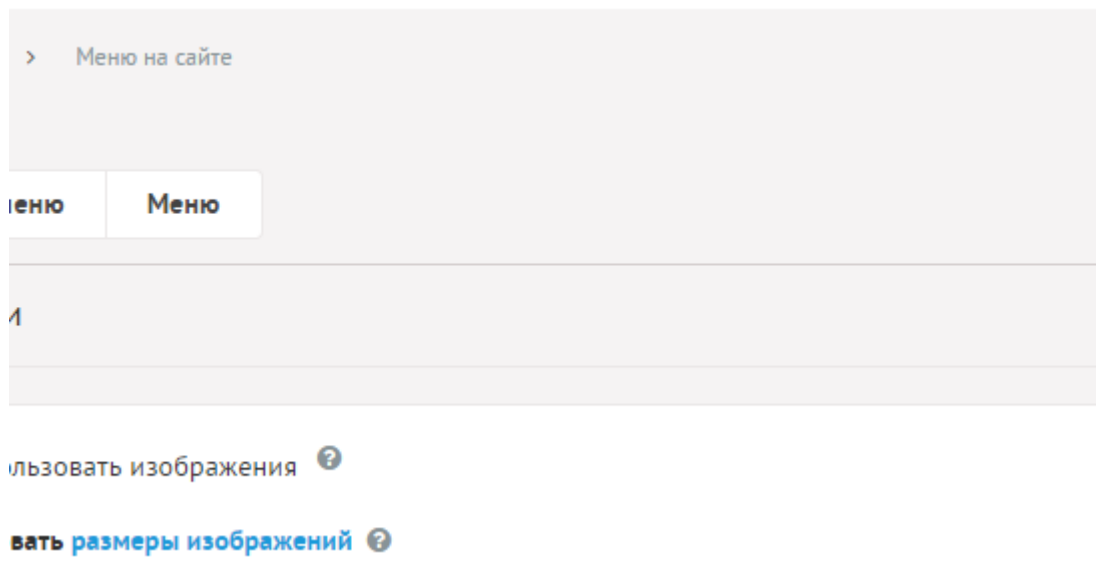
Основные:

- **ID меню** – Номер элемента в БД (Веб-мастеру и программисту).
- **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).
- **Сортировка: установить перед** – Редактирование порядка следования категории в списке. Поле доступно для редактирования только для категорий, отображаемых на

сайте.

- **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_end_after_children_номер** – текст, выводимый после вложенных пунктов каждого 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_end_after_children_номер** – текст, выводимый после вложенных пунктов

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);

— **tag_active_child_end_after_children_номер** – текст, выводимый после вложенных пунктов 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;

— **count_level** – количество выводимых уровней меню, атрибут используется при оформлении меню атрибутами (template="");

— **defer** – маркер отложенной загрузки шаблонного тега: **event** – загрузка контента только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** – загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка контента только по желанию пользователя;

— **defer_title** – текстовая строка, выводимая на месте появления загружаемого контента с помощью отложенной загрузки шаблонного тега.

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>
```

В шаблоне тега можно получить значение любого атрибута через переменную `$result["attributes"]["название атрибута"]`.

База данных

[{menu}](#) – Пункты меню

[{menu_category}](#) – Меню

[{menu_category_site_rel}](#) – Данные о том, на каких страницах сайта выводится меню

[{menu_parents}](#) – Родительские связи пунктов меню

Файлы

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

1. *modules/menu/admin/js/menu.admin.js* – редактирование пунктов меню, JS-сценарий;
2. *modules/menu/admin/menu.admin.php* – редактирование пунктов меню;
3. *modules/menu/admin/menu.admin.action.php* – обработка POST-запросов при работе с меню в административной части;
4. *modules/menu/admin/menu.admin.category.php* – редактирование категорий меню;
5. *modules/menu/admin/menu.admin.config.php* – настройки модуля;

6. *modules/menu/admin/menu.admin.inc.php* – подключение модуля к административной части других модулей;
7. *modules/menu/menu.php* – контроллер;
8. *modules/menu/menu.inc.php* – подключение модуля;
9. *modules/menu/menu.install.php* – установка модуля;
10. *modules/menu/menu.model.php* – модель;
11. *modules/menu/views/m/menu.view.show_block_topmenu.php* – шаблон меню `template=topmenu`;
12. *modules/menu/views/m/menu.view.show_level_topmenu.php* – шаблон вывода первого уровня меню, вызывается из функции `show_block` в начале файла, `template=topmenu`;
13. *modules/menu/views/m/menu.view.show_level_topmenu_2.php* – шаблон второго и последующих уровней меню, оформленного шаблоном;
14. *modules/menu/views/m/menu.view.show_level_topmenu_3.php* – шаблон второго и последующих уровней меню, оформленного шаблоном;
15. *modules/menu/views/menu.view.show_block.php* – шаблон меню, оформленного шаблоном;
16. *modules/menu/views/menu.view.show_block_leftmenu.php* – шаблон меню `template=leftmenu`;
17. *modules/menu/views/menu.view.show_block_topmenu.php* – шаблон меню `template=topmenu`;
18. *modules/menu/views/menu.view.show_level.php* – шаблон вывода первого уровня меню, вызывается из функции `show_block` в начале файла, оформленного шаблоном;
19. *modules/menu/views/menu.view.show_level_2.php* – шаблон вывода второго и последующих уровней меню, оформленного шаблоном;
20. *modules/menu/views/menu.view.show_level_leftmenu.php* – шаблон вывода первого уровня меню, вызывается из функции `show_block` в начале файла, `template=leftmenu`;
21. *modules/menu/views/menu.view.show_level_leftmenu_2.php* – шаблон второго и последующих уровней меню, оформленного шаблоном;
22. *modules/menu/views/menu.view.show_level_topmenu.php* – шаблон вывода первого уровня меню, вызывается из функции `show_block` в начале файла, `template=topmenu`;
23. *modules/menu/views/menu.view.show_level_topmenu_2.php* – шаблон второго и последующих уровней меню, оформленного шаблоном;

24. *modules/menu/views/menu.view.show_level_topmenu_3.php* – шаблон второго и последующих уровней меню, оформленного шаблоном;

25. *modules/menu/views/menu.view.show_menu.php* – шаблон меню, оформленного атрибутами тега.

Пользователи сайта

- [Подключение](#)
- [Пользователи](#)
- [Права доступа](#)
- [Регистрация](#)
- [Восстановление доступа](#)
- [Конструктор формы регистрации](#)
- [Настройки](#)
- [Шаблонные теги](#)
- [База данных](#)
- [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.

Подключение

Подключение используется для доступа к свойствам текущего пользователя и работой с текущим пользователем.

Подключаемая часть – файл *modules/users/users.inc.php*. В нем описан класс *Users_inc*. В модуле к объекту класса можно обратиться через переменную `$this->diafan->_users`. Экземпляр класса создается при первом вызове переменной.

Свойства

`var id` – номер текущего пользователя.

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->_users->id)
{
    Custom::inc('includes/403.php');
}
```

`var role_id` – роль текущего пользователя.

`var checked` – пользователь проверен по идентификационному хэшу.

Пример:

```
// в файле adm/includes/action.php
// при отправке форм производится проверка на уникальность каждого запроса
// посредством хэш-кода пользователя
if (!$this->diafan->_users->checked)
{
    echo 'Пользователь не прошел проверку.';
}
```

var **admin** – пользователь является администратором с доступом ко всем модулям.

Пример:

```
// Воспользуемся проверкой свойства admin для вывода приветствия администратору
if($this->diafan->_users->admin)
{
    echo 'Здравствуйте, администратор!';
}
```

var **errauth** – ошибка авторизации.

Пример:

```
// при ошибке авторизации содержит в себе константу данной ошибки
switch ($this->diafan->_users->errauth)
{
    case 'wrong_login_or_pass':
        echo 'Неверный логин или пароль.';
        break;

    case 'blocked_30_min':
        echo 'Вы превысили количество попыток, поэтому будете заблокированы на 30 минут';
        break;

    case 'blocked':
        echo 'Логин не активирован или заблокирован.';
        break;
}
```

var **fields = array('name', 'fio', 'mail', 'phone', 'created', 'role_id', 'lang_id', 'htmleditor', 'admin_nastr', 'start_admin', 'useradmin', 'copy_files', 'config')** – характеристики пользователя.

Поля, включенные в этот массив, можно вызывать через переменную `$this->diafan->_users`. Служебная функция `__get()` вернёт значение поля из базы данных для текущего пользователя.

Пример:


```
echo 'Здравствуйте, '.$this->diafan->_users->fio.'!  
Ваш логин: '.$this->diafan->_users->name.'  
Ваш e-mail: '.$this->diafan->_users->mail.'  
Ваш телефон: '.$this->diafan->_users->phone.'  
Вы зарегистрировались на сайте: '.date("d.m.Y H:i", $this->diafan->_users->created);
```

Методы

void **set (object \$user)** – Определяет текущего пользователя.

- object \$user: данные о текущем пользователе

Пример:

```
// авторизуем текущего посетителя сайта как пользователя с ID=5  
$user = DB::query_fetch_object("SELECT * FROM {users} WHERE id=5");  
$this->diafan->_users->set($user);
```

void **delete_session_hash (integer \$id)** – Удаляет уникальный хэш сессии.

- integer \$id: номер хэша

string **get_hash ()** – Генерируем идентификационный пользовательский хэш.

Пример:

```
// includes/action.php  
// используется для проверки уникальности запроса для данного пользователя  
$this->result["hash"] = $this->diafan->_users->get_hash();
```

boolean true **logout ()** – Очищает информацию о текущем пользователе.

Пример:

```
// includes/init.php  
// иницирует выход пользователя из системы  
if (strpos($_GET["rewrite"], "logout") !== false)  
{  
    $this->_users->logout();  
}
```

array **auth (array \$form_values)** – Проверяет авторизован ли пользователь.

- array \$form_values: массив с данными для авторизации - логин и пароль

Пример:

```
// includes/init.php  
// проверка данных при авторизации пользователя  
if (! empty($_POST['action']) && $_POST['action'] == 'auth')  
{  
    $this->_users->auth($_POST);  
}
```

```
}
```

boolean **auth_loginza (array \$form_values)** – Проверяет авторизован ли пользователь.

- array \$form_values: массив с данными для авторизации - логин и пароль

Пример:

```
// includes/init.php
// проверка данных при авторизации пользователя с возможностью авторизации через
сервис Loginza
if (! empty($_POST['action']) && $_POST['action'] == 'auth')
{
    $this->_users->auth($_POST);
}
elseif (! empty($_POST['token']))
{
    $this->_users->auth_loginza();
}
```

void **create_avatar (string \$name, string \$file)** – Загружает аватар.

- string \$name: логин пользователь
- string \$file: файл аватара

Пример:

```
// сгенерирует аватар для пользователя с логином alex
// из файла http://site.ru/userfiles/alex_avatar.jpg
$name_user = 'alex';
$path_to_img = ABSOLUTE_PATH.USERFILES.'/alex_avatar.jpg';
$this->diafan->_users->create_avatar($name_user, $path_to_img);
```

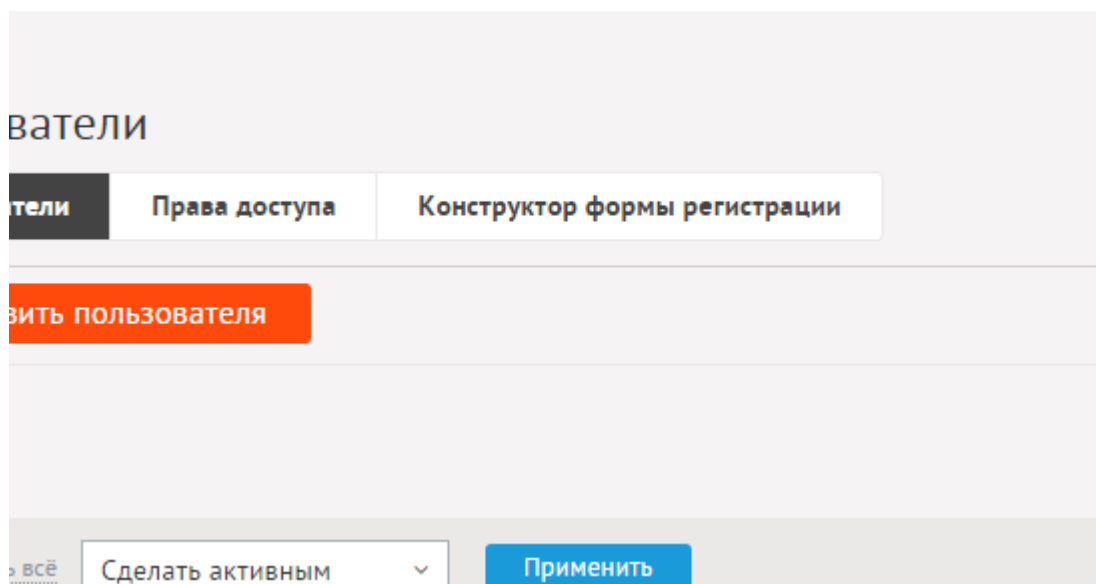
boolean **roles (string \$action, [string \$module_name = "], [array \$roles = array()], [string \$type = 'admin'])** – Проверяет есть ли права у пользователя на действие для модуля.

- string \$action: действие
- string \$module_name: модуль
- array \$roles: права пользователя
- string \$type: часть сайта административная/пользовательская

Пример:

```
// проверка прав пользователя на редактирование категорий товаров
if (! $this->diafan->_users->roles("edit", "shop/category"))
{
    echo "Нет доступа к модулю.";
}
```

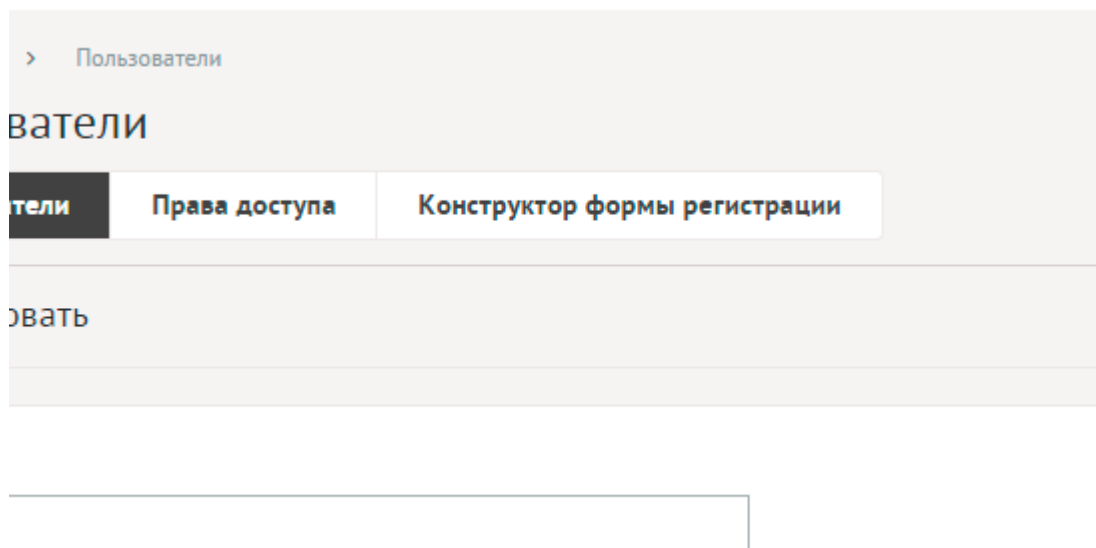
Пользователи



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

Администратор не может удалить, заблокировать самого себя или изменить себе права доступа. Пользователь с правами администратора может авторизоваться на сайте только по секретному адресу [панели управления сайтом](#).

Пользователей можно искать по ФИО.



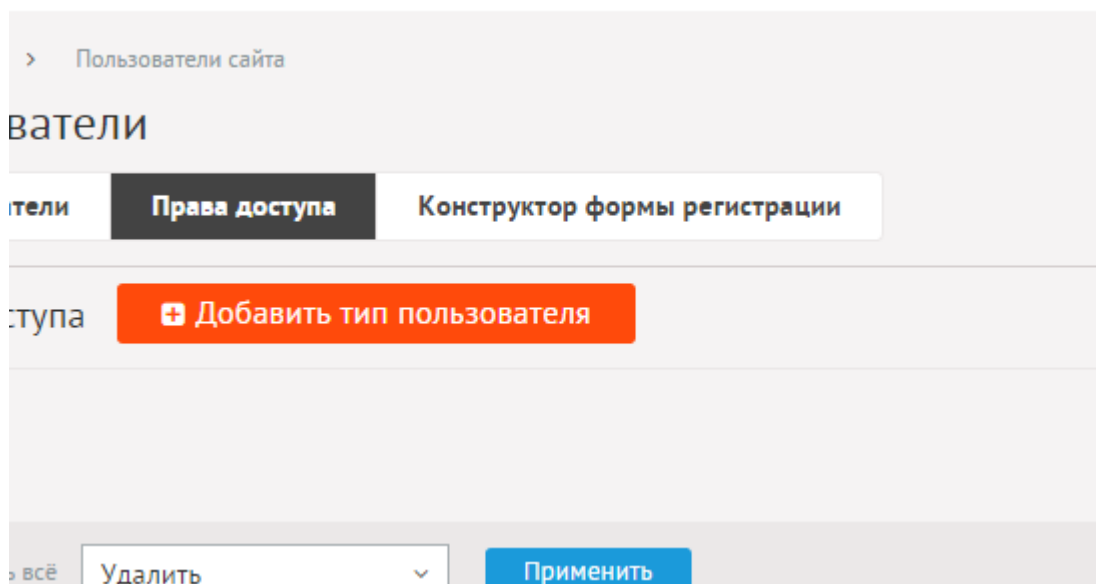
Пользователи имеют следующие характеристики:

Основные:

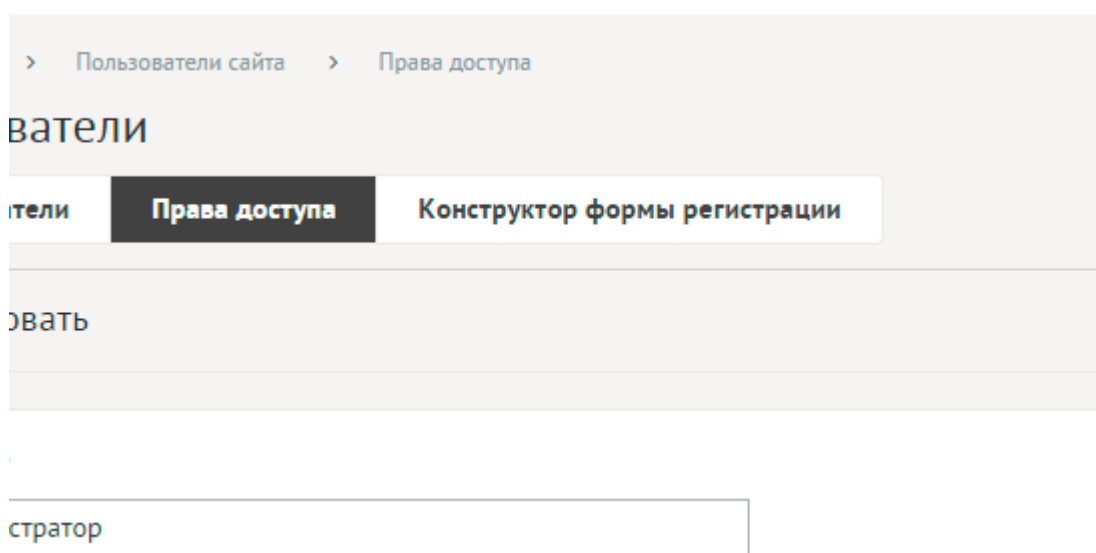
- **ФИО** – Свободное информационное текстовое поле.
- **Дата регистрации** – В формате дд.мм.гггг чч:мм, при регистрации устанавливается текущая.
- **Активен** – Если не отмечена, пользователь не сможет авторизоваться на сайте

- **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».
- **Активность пользователя.**
- **Активность.**
- **Дополнительно.**
- **URL на страницу в соц. сети** – Заполняется при авторизации через соц. сеть. Служит идентификатором, с помощью которого пользователь может зайти в аккаунт.
- **Язык пользовательской части сайта по умолчанию** – 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.
- **Настройки для административной части.**
- **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».
- **Панель быстрого редактирования** – Настраивает вывод в пользовательской части сайта в шапке панели, с помощью которой можно редактировать контент прямо на сайте. Возможные значения:
 - отключена,
 - включена,
 - только панель без режима редактирования.
- **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.

Права доступа



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



Пользователи с любыми правами доступа в административную часть сайта могут авторизоваться на сайте только по секретному адресу [панели управления сайтом](#).

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;
    }
}
```

- **Сортировка: установить перед** – Редактирование порядка следования типа пользователя в списке.

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.

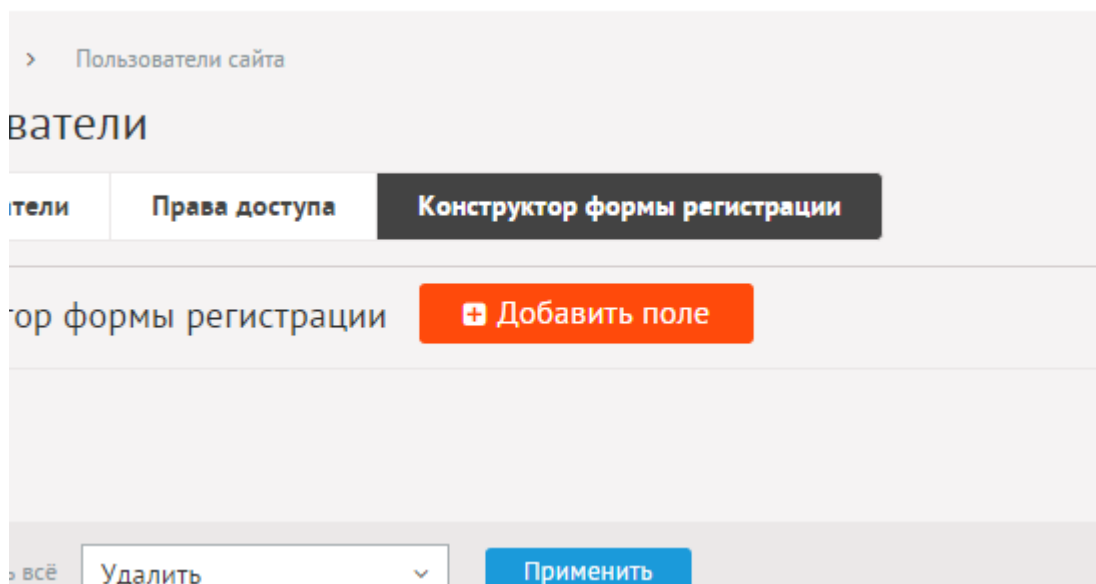
Восстановление доступа

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.

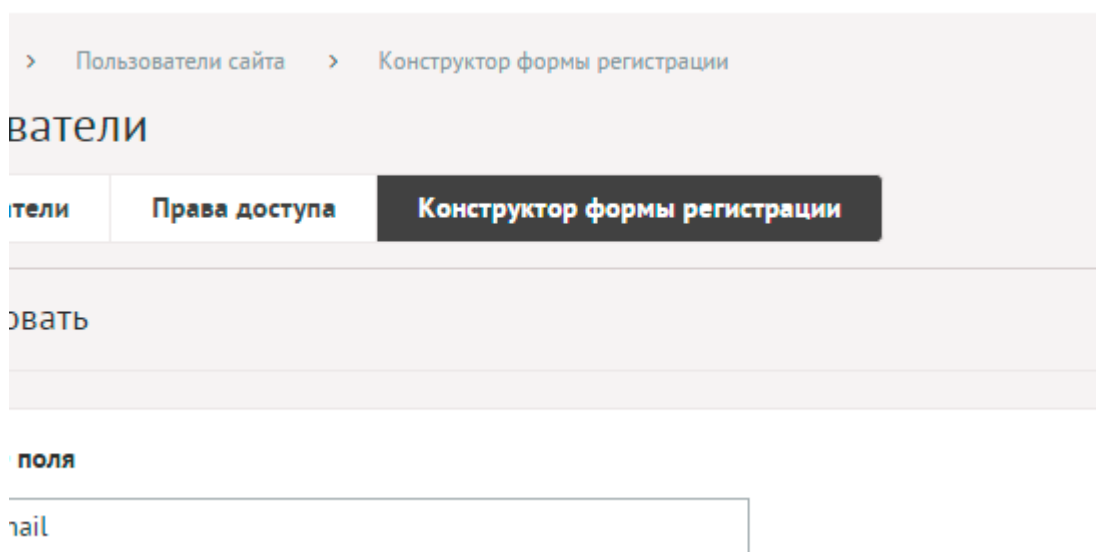
Восстановление доступа осуществляются с помощью технологии Ajax, то есть без перезагрузки всей страницы.

Конструктор формы регистрации



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

Конструктор – это дополнительные поля формы регистрации.



Дополнительные поля имеют следующие параметры.

Основные:

- **Название поля** – 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,
 - телефон,
 - 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».
- **Обязательно для заполнения.**
- **Выводить на странице пользователя.**
- **Выводить в форме регистрации.**
- **Выводить в форме редактирования данных.**

- **Только для пользователей** – 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.

Настройки

> Пользователи сайта

Пользователи

Пользователи	Права доступа	Конструктор формы регистрации
1		

активации пользователя

ссылке, высланной на e-mail

- **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 удачно зарегистрированы. Для активации аккаунта пройдите по ссылке, высланной на ваш e-mail.

- **Скрывать форму после успешной регистрации** – Позволяет скрывать форму после успешной регистрации.
- **Формат даты** – Позволяет настроить отображение даты в модуле. Возможные значения:
 - 01.05.2016,
 - 01.05.2016 14:45,
 - 1 мая 2016 г.,

- 1 мая,
- 1 мая 2016, 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».
- **Уведомления.**
- **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 перейдите по ссылке
 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.

• **Тема письма пользователю при активации аккаунта администратором** – Можно добавлять:

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

Параметр имеет разные значения для разных языковых версий сайта.

Пример:

Ваш аккаунт на сайте %title (%url) активирован администратором

• **Сообщение пользователю при активации аккаунта администратором** – Можно добавлять:

- %title – название сайта,
- %url – адрес сайта (например, site.ru),
- %fio – ФИО пользователя,
- %login – логин пользователя,
- %email – e-mail пользователя.

Параметр имеет разные значения для разных языковых версий сайта.

Пример:

Здравствуйте, %fio!
 Ваш аккаунт на сайте %title (%url) активирован администратором.
 Логин: %login

- **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 перейдите по ссылке.

- **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

- **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».

Шаблонные теги

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

show_block – выводит статистику пользователей на сайте.

Атрибуты:

— **defer** – маркер отложенной загрузки шаблонного тега: **event** – загрузка контента только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** – загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка контента только по желанию пользователя;

— **defer_title** – текстовая строка, выводимая на месте появления загружаемого контента с помощью отложенной загрузки шаблонного тега;

— **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.»

В шаблоне тега можно получить значение любого атрибута через переменную `$result["attributes"]["название атрибута"]`.

show_login module="registration" – выводит форму авторизации или приветствие и ссылки на редактирование данных и выход, если пользователь авторизован.

Атрибуты:

- **defer** – маркер отложенной загрузки шаблонного тега: **event** – загрузка контента только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** – загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка контента только по желанию пользователя;
- **defer_title** – текстовая строка, выводимая на месте появления загружаемого контента с помощью отложенной загрузки шаблонного тега;
- **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

В шаблоне тега можно получить значение любого атрибута через переменную `$result["attributes"]["название атрибута"]`.

База данных

[{users}](#) – Пользователи

[{users_actlink}](#) – Код активации аккаунта

[{users_param}](#) – Дополнительные поля с данными о пользователях

[{users_param_element}](#) – Значения дополнительных полей с данными о пользователях

[{users_param_role_rel}](#) – Связи дополнительных полей с данными о пользователях и типов пользователей

[{users_param_select}](#) – Варианты значений дополнительных полей с данными о пользователях для типа список

[{users_role}](#) – Типы пользователей

[{users_role_perm}](#) – Права типов пользователей

Файлы

1. *modules/registration/js/registration.form.js* – JS-сценарий модуля «Регистрация»;

2. *modules/registration/registration.php* – контроллер модуля «Регистрация»;
3. *modules/registration/registration.action.php* – обработка запроса при регистрации пользователя;
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.action.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/js/users.admin.edit.js* – редактирование пользователей, JS-сценарий;

- 22. *modules/users/admin/js/users.admin.role.js* – редактирование типов пользователей, JS-сценарий;
- 23. *modules/users/admin/users.admin.php* – редактирование пользователей сайта;
- 24. *modules/users/admin/users.admin.config.php* – настройки модуля;
- 25. *modules/users/admin/users.admin.inc.php* – подключение модуля к административной части других модулей;
- 26. *modules/users/admin/users.admin.param.php* – конструктор формы регистрации;
- 27. *modules/users/admin/users.admin.role.php* – редактирование типов пользователей;
- 28. *modules/users/users.php* – контроллер модуля «Пользователи на сайте»;
- 29. *modules/users/users.inc.php* – подключение модуля;
- 30. *modules/users/users.install.php* – установка модуля;
- 31. *modules/users/users.model.php* – модель модуля «Пользователи на сайте»;
- 32. *modules/users/views/users.view.show_block.php* – шаблон статистики пользователей на сайте;
- 33. *modules/usersettings/js/usersettings.form.js* – JS-сценарий модуля «Настройки аккаунта»;
- 34. *modules/usersettings/usersettings.php* – контроллер модуля «Настройки аккаунта»;
- 35. *modules/usersettings/usersettings.action.php* – обработка запроса при изменении данных о пользователе;
- 36. *modules/usersettings/usersettings.model.php* – модель модуля «Настройки аккаунта»;
- 37. *modules/usersettings/views/usersettings.view.attachments.php* – шаблон прикрепленных файлов в настройках аккаунта;
- 38. *modules/usersettings/views/usersettings.view.avatar.php* – шаблон аватара;
- 39. *modules/usersettings/views/usersettings.view.form.php* – шаблон формы настроек аккаунта;
- 40. *modules/usersettings/views/usersettings.view.images.php* – шаблон прикрепленных изображений;
- 41. *modules/usersettings/views/usersettings.view.show_param.php* – шаблон дополнительных в настройках аккаунта.

Темы и дизайн

- [Как создать тему](#)
- [Подключение](#)
- [Файлы](#)

Как правило, на сайтах не используется исходная CMS с исходным дизайном. Веб-мастера интегрируют свои шаблоны дизайна, вносят свои доработки в коды файлов модулей (мы называем это «кастомизацией», от англ. custom - сделанный на заказ, пользовательский, переработанный). Веб-мастер может очень долго перерабатывать DIAFAN.CMS, меняя модули «под себя», затрагивая много файлов системы в разных папках.

Для удобства систематизации и администрирования таких изменений в DIAFAN.CMS существуют **Темы**.

Внимание! **Тема** – набор пользовательских файлов, отличающихся от стандартных файлов DIAFAN.CMS, вынесенных в отдельную папку *custom*, с сохранением иерархии.

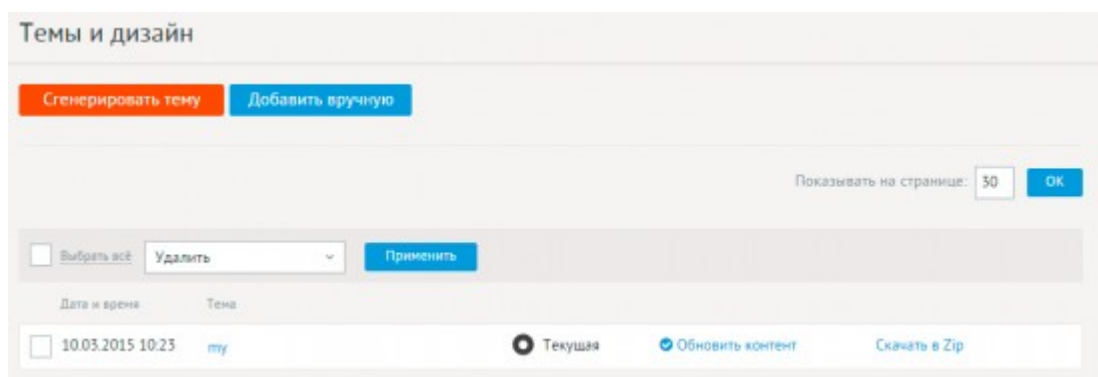
Темы могут потребоваться для того, чтобы:

- перенести доработки на другой сайт на DIAFAN.CMS или поделиться с сообществом;
- быстро и удобно менять дизайн сайта;
- защитить свои доработки от обновлений.

Внимание! Обратите внимание на [готовые темы дизайна для DIAFAN.CMS](#)

Как создать тему

- При [обновлении системы](#) и получении новых файлов DIAFAN.CMS сама определяет измененные Вами файлы, сравнивая их с сохраненными копиями в точке возврата. В ходе обновления тема сгенерируется автоматически и все «кастомные» файлы перенесутся в папку *custom*.
- После доработки модулей и/или внедрения своего дизайна, пользователь может самостоятельно зайти в модуль «Темы и дизайн», нажать «Сгенерировать тему», и все «кастомные» файлы автоматически перенесутся в папку *custom*.



- При переносе темы с другого сайта, нужно сначала создать и применить новую тему, а затем скопировать файлы переносимой темы в папку темы. Она будет создана в папке *custom*. При ручном переносе проследите, чтобы сохранилась структура папок,

ВХОДЯЩИХ В СОСТАВ ТЕМЫ.

Внимание! **Помните!** Файлы текущей темы имеют приоритет над аналогичными файлами системы! Поэтому если Вы, [внедряя свой дизайн](#), правили системный файл стандартного демо-шаблона *themes/site.php*, после создания темы этот файл переедет в папку *custom/theme-name/themes/site.php*, а на месте *themes/site.php* восстановится исходный файл DIAFAN.CMS! Это же касается всех остальных файлов, которые Вы трогали, в том числе и *css/style.css*, который после генерации темы будет иметь адрес *custom/theme-name/css/style.css* и т.д.

Пример:

Веб-мастер ранее уже вносил изменения в тему, и/или применял обновления. В результате на сайте несколько тем, одна из которых сейчас используется. Например, эта текущая тема называется "mytheme", соответственно её файлы хранятся в папке *custom/mytheme*.

Если веб-мастеру необходимо исправить внешний вид списка новостей, необходимо править файл *custom/mytheme/modules/news/views/news.view.list.php*. Изменения отобразятся на сайте, потому что это текущая тема.

Частая ошибка, которую совершают веб-мастера, забывая, что у них применена тема. Вместо файла текущей темы правят «обманчиво похожий» исходный системный файл *modules/news/views/news.view.list.php*. Все изменения из этого файла не будут отображены на сайте, потому что приоритет показа у аналогичных файлов из папки текущей темы *custom/mytheme*.

Итак, если Вы столкнетесь с тем, что изменения файлов темы не отображаются на сайте – проверьте какую тему редактируете, свою текущую или базовую «из коробки».

Вы также можете применить сразу несколько тем. Тогда они будут использоваться на сайте в порядке установки. В приоритете будут последние установки. В списке тем примененные темы выводятся выше остальных, в порядке применения. Сначала те, что применялись раньше. Затем те, что применялись позже.

Пример:

На сайте применено две темы. Сначала тема *my*, затем тема *my1*. В обеих темах есть файл *modules/news/news.php*. На сайте будет использоваться файл из папки *my1*, так как эта тема применена позже.

Если же какие-то файлы есть только в теме *my*, то они будут использованы.

Сколько тем можно создавать для одного сайта?

Сколько угодно. Переключаться между ними очень просто. Нужно только применить нужную тему и отключить прежнюю тему. Чаще всего темы используются для быстрого изменения дизайна, при [загрузке шаблонов](#) или переносе сайтов.

Как перенести тему с одного сайта на другой?

Сначала нужно сохранить все «кастомные» файлы, сгенерировав тему. Затем просто скачать архив темы с одного сайта и создать тему на другом сайте, добавив скачанный архив в поле «Импорт темы».

Может ли тема быть пустой?

Может, хотя практического смысла это не имеет. Если создать новую тему и ничего больше с ней не делать, папка темы в папке *custom* появится, создание темы будет «засчитано» в системе. Только никакого влияния на отображение сайта это, разумеется, не окажет. Скачать пустую тему в виде ZIP-архива тоже нельзя.

Демонстрационный контент для темы

Если Вы создаете шаблон для [сайта дополнений](#), Вы можете добавить в свою тему демонстрационный контент, который поможет лучше представить назначение и область применения шаблона. В DIAFAN.CMS есть автогенератор демонстрационного контента. Для того, чтобы добавить демонстрационный контент нужно:

- заполнить сайт так, как его должен увидеть покупатель шаблона;
- удалить (в том числе из корзины) все лишние данные, отключить лишние модули;
- нажать ссылку «Добавить в текущую тему демонстрационный контент».

После этого в Вашей текущей примененной теме появятся файлы установки контента для модулей. Контент модулей будет установлен, если загрузить тему во время установки DIAFAN.CMS или при переустановки модуля в интерфейсе «Установка/удаление модулей».

Подключение

Подключаемая часть – файл *modules/custom/custom.inc.php*. В нем описан класс *Custom_inc*. В модуле к объекту класса можно обратиться через переменную `$this->diafan->_custom`. Экземпляр класса создается при первом вызове переменной.

Методы

array **generate ()** – Генерирует тему из кастомизированных файлов.

Пример:

```
// сгенерирует тему
$result = $this->diafan->_custom->generate();
if(! $result)
{
    echo 'Тема не сгенерирована. Нет кастомизированных файлов.';
}
else
{
    if($result["custom"])
    {
        echo 'Кастомизированные файлы: '.implode(' ', $result["custom"]);
    }
    if($result["return"])
```

```

{
    echo 'Восстановленные файлы: '.implode(' ', $result["return"]);
}
}
// выведет: перечень файлов

```

boolean **set (mixed \$array, boolean \$enable, [boolean \$sql = false])** – Изменяет состояние темы.

- mixed \$array: название темы или массив названий тем
- boolean \$enable: активирует тему
- boolean \$sql: выполняет дополнительные запросы к базе данных

boolean **import (string \$file_path, string \$name)** – Импортирует тему.

- string \$file_path: архивный файл темы
- string \$name: название темы

void **query (string \$name, [boolean \$install = true])** – Исполнение SQL-запросов в файле install.sql или uninstall.sql.

- string \$name: название темы
- boolean \$install: определяет файл запросов: install или uninstall

array **get_modules ([mixed \$names = false])** – Получает список всех модулей которые можно установить.

- mixed \$names: темы, для которых определяются модули (по умолчанию все активные темы)

boolean **set_modules (mixed \$modules, boolean \$enable, [mixed \$names = false])** – Установка/удаление модулей.

- mixed \$modules: название модуля или массив названий модулей
- boolean \$enable: маркер установки/удаления модулей
- mixed \$names: не активные темы, которые необходимо временно подключить для установки/удаления определенных в их коде модулей

Файлы

1. *modules/custom/admin/custom.admin.php* – темы;
2. *modules/custom/admin/custom.admin.demo.php* – демо-данные для темы;
3. *modules/custom/admin/js/custom.admin.edit.js* – редактирование темы, JS-сценарий;
4. *modules/custom/admin/js/custom.admin.js* – темы, JS-сценарий;
5. *modules/custom/custom.export.php* – экспорт темы;
6. *modules/custom/custom.inc.php* – подключение модуля;

7. *modules/custom/custom.install.php* – установка модуля.

Обновление CMS

- [Автообновление](#)
- [Подключение](#)
- [Файлы](#)

Модуль «Обновление CMS» обязателен для установки. Он относится к CMS «Управление» и является частью ядра DIAFAN.CMS. Имеет только административную часть.

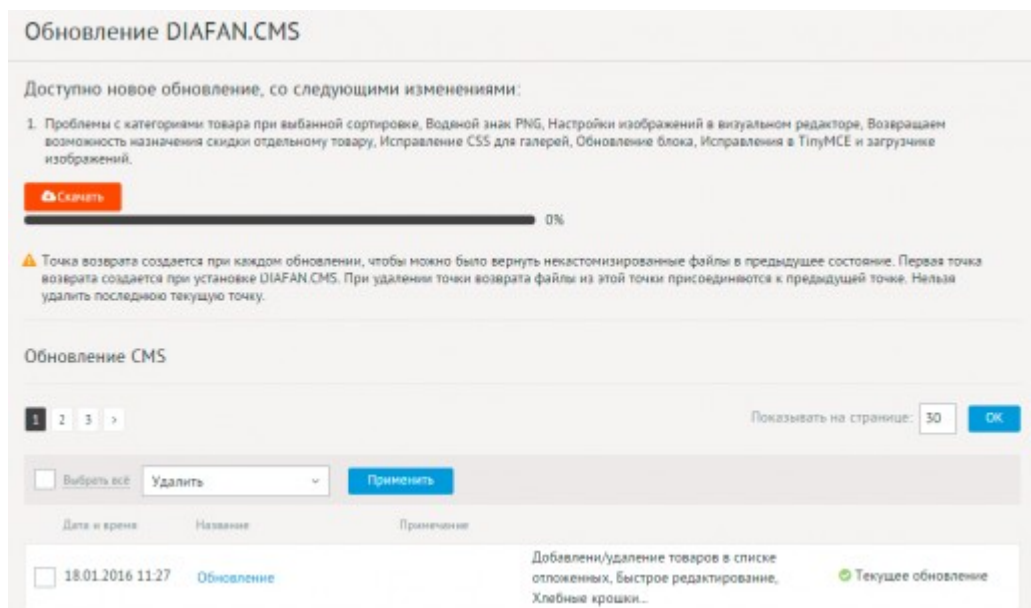
Для корректной работы модуля необходимо PHP-расширение ZipArchive.

Автообновление

Модуль позволяет получать обновления для Вашей DIAFAN.CMS с нашего сервера.

При первой установке DIAFAN.CMS создается так называемая «точка возврата», которая содержит резервные копии всех установленных файлов «из коробки». В дальнейшем при получении обновлений, DIAFAN.CMS создает точки возврата и сохраняет отдельно обновленные версии файлов.

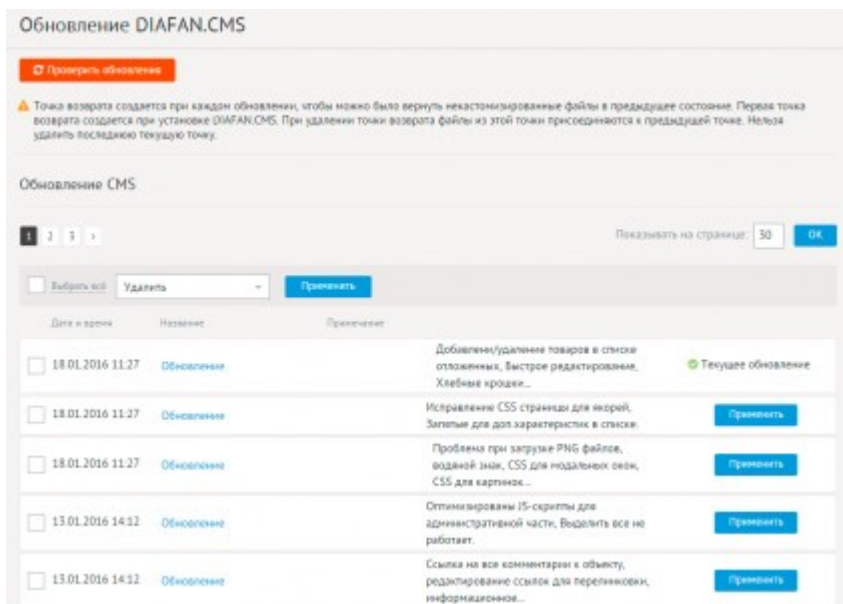
Обновления для DIAFAN.CMS проверяются только по Вашему запросу! Необходимо зайти в модуль «Обновление CMS» и нажав кнопку «Проверить обновление». Если для Вашей версии DIAFAN.CMS есть изменения, Вы увидите список обновлений, доступных для скачивания, и под списком будет кнопка «Скачать».



После клика по ней обновления начнут последовательно скачиваться. Если прервать загрузку до завершения, при следующей проверке список обновлений будет скорректирован. Продолжить прерванное скачивание можно аналогичным способом – с помощью кнопки «Скачать» под списком. В обновлении могут участвовать все файлы DIAFAN.CMS, в том числе шаблоны, файлы стилей и изображения. При этом пропустить какой-либо из пакетов невозможно, система автоматически проверяет целостность каждого обновления.

При каждом скачивании обновлений точки возврата создаются автоматически. После

скачивания обновления необходимо нажать кнопку «Применить» (она показывается на экране после последнего скачанного обновления). Применены будут все обновления от текущей точки возврата до последнего выполненного обновления.



Как работают обновления?

1. [Сгенерируется тема](#). Что это значит? Все полученные обновленные файлы с нашего сервера сравниваются с текущими файлами всей системы в текущей точке возврата. Если DIAFAN.CMS находит, что какие-то текущие файлы системы не соответствуют полученным из обновления, они будут перемещены в папку *custom* в текущую тему. Если текущей темы нет, то будет создана новая.
2. На место перемещенных файлов будут записаны файлы, полученные из обновления. На сайте это никак не отобразится, потому что файлы из текущей темы обладают более высоким приоритетом.
3. Если в обновлении предусмотрено изменение структуры базы данных, то оно будет произведено.

Что делать, если функционал обновления не устраивает?

Можно вернуть всё обратно в один клик. Для этого достаточно зайти в модуль «Обновление CMS» и нажать «Применить» напротив любой предыдущей точки возврата. Откат к сохраненному состоянию касается также базы данных. Если с ней были произведены какие-то манипуляции, после применения точки возврата структура БД вернется к прежнему состоянию.

Внимание! Восстановлению с помощью точек возврата подлежит только структура базы данных, не содержимое! Для полной резервной копии нужно делать сначала экспорт, а потом [импорт БД](#).

Можно ли удалять точки возврата?

Да, можно. При удалении точки возврата ее файлы присоединяются к следующей точке. Более того, удалять лишние точки возврата рекомендуется, потому что это ускорит

генерирование темы. А также, не рекомендуется увлекаться слишком частым созданием точек возврата. Ведь система инспектирует все предыдущие точки, и чем их меньше, тем быстрее происходит генерирование.

Текущую точку удалить нельзя, также как и точки возврата скачанных, но еще не примененных обновлений.

Как посмотреть детализацию точек возврата?

Каждую точку возврата можно раскрыть, кликнув по ее названию. После этого будет показан полный список файлов, которые входят в точку возврата. Кроме того, будет показан список файлов, отличающихся от текущей версии сайта.

Обновления доступны только для сайтов с активированными лицензиями DIAFAN.CMS.

Подключение

Подключаемая часть – файл *modules/update/update.inc.php*. В нем описан класс *Update_inc*. В модуле к объекту класса можно обратиться через переменную `$this->diafan->_update`. Экземпляр класса создается при первом вызове переменной.

Свойства

`var folders = array('adm', 'css', 'img', 'themes', 'modules', 'includes', 'plugins', 'js')` – папки и файлы, индексируемые для точек возврата.

`var exclude = array('adm/htmleditor', 'includes/custom.php')` – папки и файлы, не индексируемые для точек возврата.

Методы

`void first_return ()` – Добавляет первую точку возврата.

Пример:

```
// при установке DIAFAN.CMS
$this->diafan->_update->first_return();
```

`array get_files (integer $id)` – Получает обновленные файлы точки с содержимым.

- integer \$id: идентификатор точки

Пример:

```
// получает файлы точки возврата ID=3
$rows = $this->diafan->_update->get_files(3);
print_r($rows);
/* выведит:
Array
(
    [modules/news/news.model.php] => <?php
    ...
```



```

class News_model extends Model
{
...
[modules/news/news.php] => <?php
...
class News extends Controller
{
...
...
) */

```

array **get_all_files (integer \$id)** – Получает все файлы DIAFAN.CMS в точке с содержимым.

- integer \$id: идентификатор точки

Пример:

```

// получает все файлы DIAFAN.CMS, актуальные для точки возврата ID=3
$rows = $this->diafan->_update->get_all_files(3);
print_r($rows);
// выведет файлы в формате как функция get_files()

```

Файлы

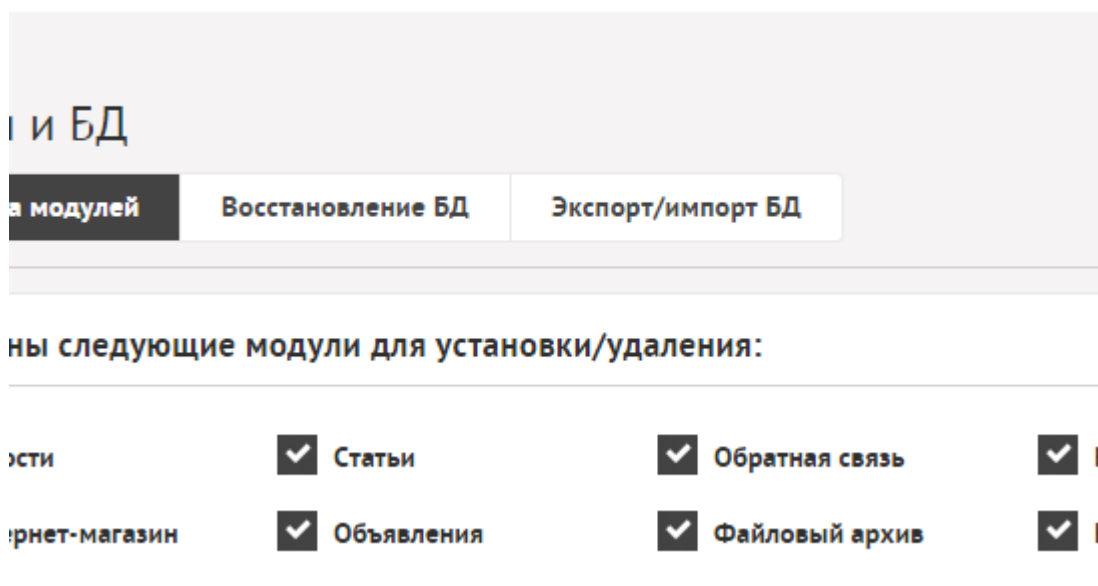
1. *modules/update/admin/js/update.admin.count.js* – количество доступных обновлений, JS-сценарий;
2. *modules/update/admin/js/update.admin.js* – обновление, JS-сценарий;
3. *modules/update/admin/update.admin.php* – точки возврата;
4. *modules/update/admin/update.admin.action.php* – точки возврата;
5. *modules/update/admin/update.admin.count.php* – количество доступных обновлений для меню административной панели;
6. *modules/update/update.inc.php* – подключение модуля;
7. *modules/update/update.install.php* – установка модуля.

Модули и БД

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

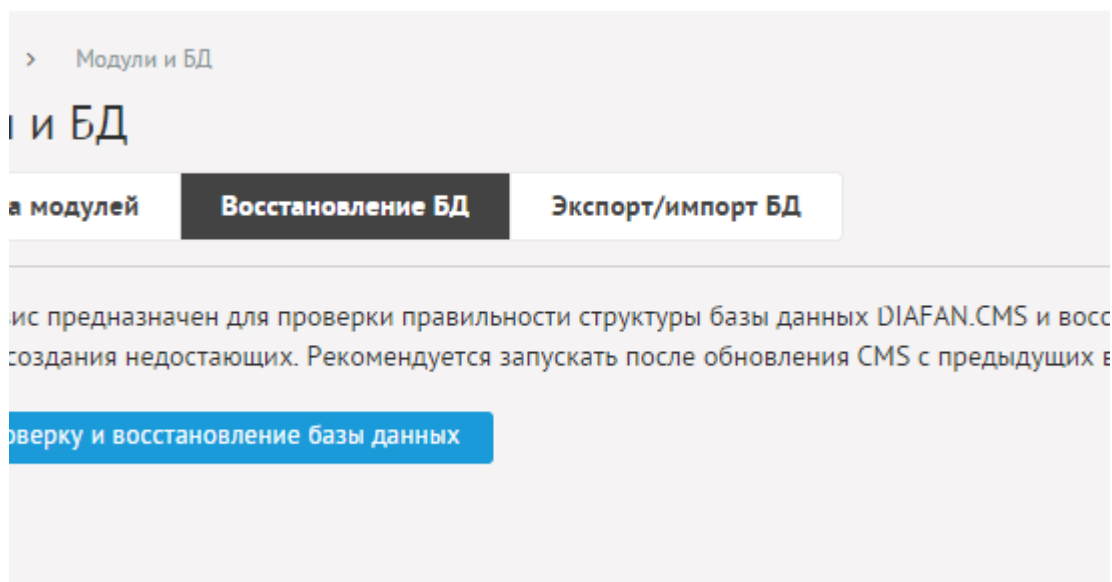
Модуль относится к части CMS «Настройки», является частью ядра DIAFAN.CMS и обязателен для установки. Имеет только административную часть.

Установка модулей



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

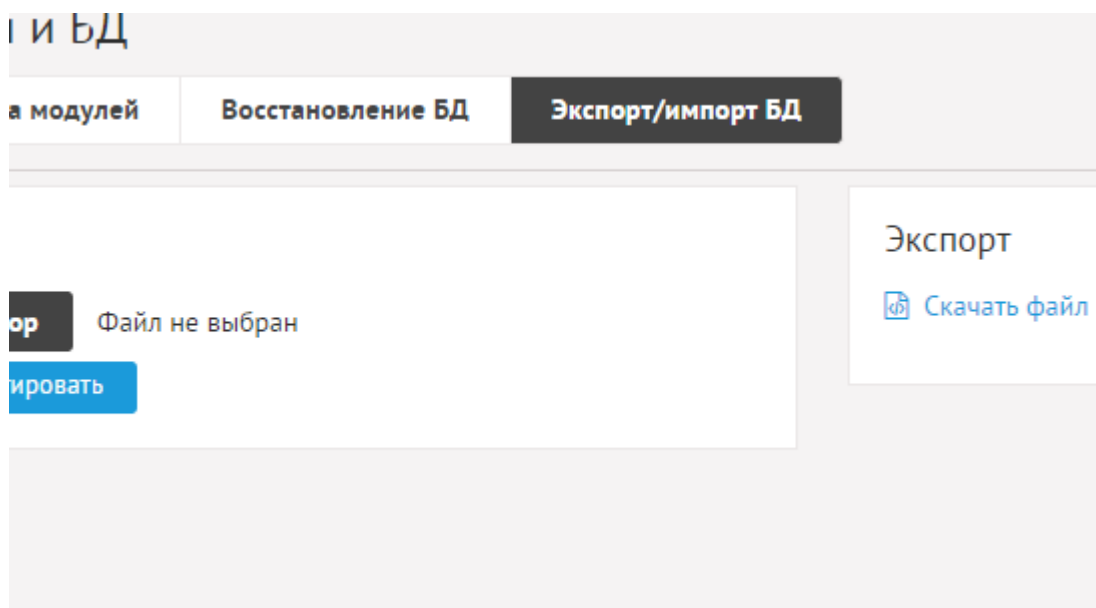


Если обновление на новую минорную версию произведено некорректно или внесены непоправимые изменения в структуру таблиц базы данных, можно воспользоваться инструментом «Восстановление базы данных».

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.

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

Экспорт/импорт базы данных



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. *modules/service/admin/service.admin.php* – установка/удаление модулей;
2. *modules/service/admin/service.admin.db.php* – импорт/экспорт базы данных;
3. *modules/service/admin/service.admin.repair.php* – восстановление базы данных;
4. *modules/service/service.export.php* – экспорт БД;
5. *modules/service/service.install.php* – установка модуля.

Параметры сайта

- [База данных](#)
- [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.
- **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.
- **Пользователь базы данных** – Данные обычно предоставляются хостингом при регистрации или создается база данных в панели управления хостингом.
- **Пароль для базы данных** – Данные обычно предоставляются хостингом при регистрации или создается база данных в панели управления хостингом.
- **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.
- **Кодировка базы данных** – DIAFAN.CMS работает с базой данных в кодировке UTF8. Изменить параметр можно в случае индивидуальной настройки системы.
- **Папка для хранения пользовательских файлов.** – Имя папки, где будут храниться все загружаемые файлы для контента сайта. По умолчанию все пользовательские файлы хранятся в папке *userfls*. (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!)
- **Использовать отдельный шаблон мобильной версии (при наличии)** – Если отмечено, то CMS будет автоматически определять устройство, с которого зашли на сайт и если это мобильное устройство, то автоматически будет загружаться дополнительный шаблон дизайна *themes/m/site.php*.
- **Имя мобильной версии в URL-адресе** – Название, используемое в URL-адресе, в качестве адреса мобильной версии. Допустимо использование латиницы в нижнем регистре, а также символов тире и нижнего подчеркивания. Например, *http://site.ru/*

m/ или <http://site.ru/mobile/>. Изменение параметра означает изменение URL-адреса мобильной версии. При изменении параметра следует скорректировать содержание файла robots.txt по необходимости. (Веб-мастеру и программисту. Не меняйте этот параметр, если не уверены в результате!) Параметр появляется, если отмечена опция «Использовать отдельный шаблон мобильной версии (при наличии)».

- **Использовать имя мобильной версии в качестве поддомена** – Если отмечено, то название мобильной версии будет использоваться в качестве поддомена. Например, <http://m.site.ru/> или <http://mobile.site.ru/>. Изменение параметра означает изменение URL-адреса мобильной версии. Возможно Вам потребуется скорректировать файл robots.txt. ВАЖНО: требуется внесение соответствующих "CNAME" или "A" записей в dns-зону домена, а также изменение настроек веб-сервера. Например, для Apache параметр "ServerAlias", для NGINX параметр "server_name". Прежде, чем изменять параметр настройки, убедитесь, что имя мобильной версии не совпадает ни с одной из частей доменного имени. Например, для URL-адреса <http://site.ru/> в качестве имени мобильной версии нельзя использовать: *site* и *ru*. (Веб-мастеру и программисту. Не меняйте этот параметр, если не уверены в результате!) Параметр появляется, если отмечена опция «Использовать отдельный шаблон мобильной версии (при наличии)».
- **Запретить загружать сайт во frame** – Если не отмечено, то сайт можно будет вставлять во frame. 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:

|a|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|A|B|B|Г|Д|E|Ё|Ж|З|И|Й|K|Л|M|N|O|P|R|S|T|U|F|X|Ц|Ч|Ш|Щ|Ы|Э|Ю|Я

Пример:

-|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|Y|O|Z|H|Z|I|Y|K|L|M|N|O|P|R|S|T|U|F|KH|TS|CH|SH|SCH|Y|E|YU|YA

- **API-ключ сервиса Яндекс Переводчик**
[Получить.](#)
- **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/](http://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** – Обязательно используйте исходящую SMTP-авторизацию, иначе письма-уведомления с сайта могут блокироваться большинством спам-фильтров.
- **SMTP-хост (например, tls://smtp.mail.ru)** – Параметр появляется, если отмечена опция «Использовать SMTP-авторизацию при отправке почты с сайта».
- **SMTP-логин (например, ivanov@mail.ru)** – 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 (например, 465 или 587)** – В большинстве случаев можно не указывать. Если используется протокол SSL, то чаще всего необходимо указывать SMTP-порт 465. Если используется протокол TLS, то чаще всего необходимо указывать SMTP-порт 587. 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 [SMSC](#).
- **Ключ** – Данные из настроек сервиса SMSC. Параметр появляется, если отмечена опция «Подключить SMS-уведомления(требуется регистрация)».
- **ID** – Данные из настроек сервиса SMSC. Параметр появляется, если отмечена опция «Подключить SMS-уведомления(требуется регистрация)».

После регистрации в системе [Byte Hand](#) **Ключ** и **ID** можно найти на странице «Настройки» (в меню «Аккаунт» – «Настройки») в аккаунте пользователя системы Byte Hand.

Настройки ID [REDACTED] Ключ [REDACTED]

Данные учетной записи

Логин [REDACTED] [Изменить пароль...](#)

Граница баланса для уведомлений, руб
100

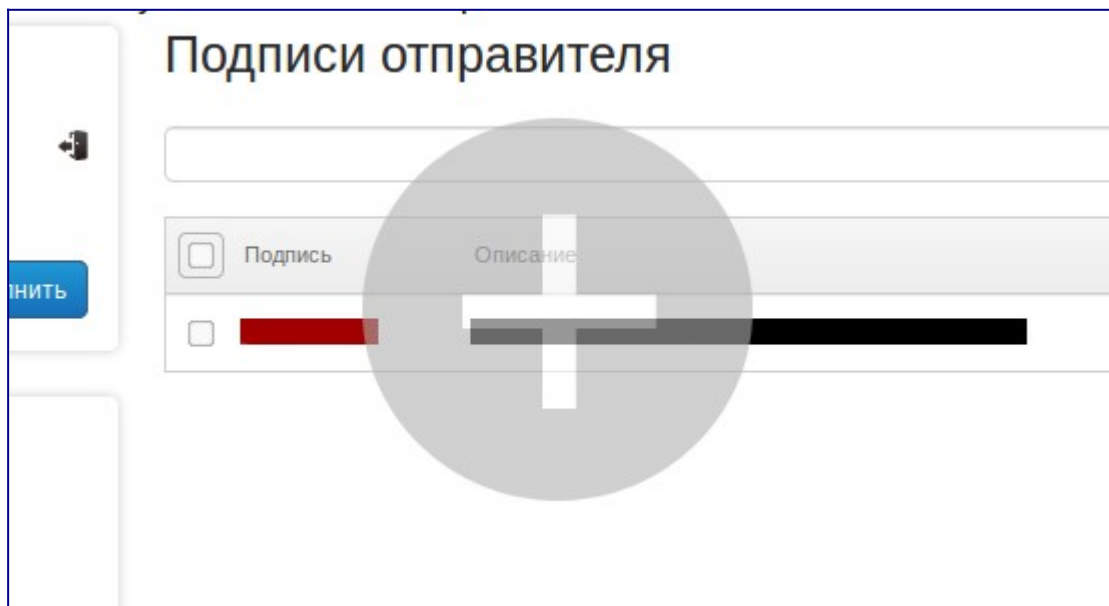
Часовой пояс
GMT+03:00 Москва +00

Контактные данные

При достижении указанной суммы уведомление на E-mail придет чаще чем раз в день.

- **Подпись** – Данные из настроек сервиса SMSC. Параметр появляется, если отмечена опция «Подключить SMS-уведомления(требуется регистрация)».

Подпись необходимо создавать в аккаунте системы Byte Hand (в меню «Рассылки» – «Подпись отправителя»). Отправка SMS будет доступна, когда подпись, указанная в параметрах, пройдет модерацию и будет иметь статус «Доступна».



- **Таймзона** – Часовой пояс, [список часовых поясов](#). По умолчанию: Europe/Moscow
- **Источник загрузки JS-библиотек** – Вариант источника загрузки библиотек JavaScript. Влияет на скорость загрузки страниц сайта. (Веб-мастеру и программисту. Не меняйте этот параметр, если не уверены в результате!) Возможные значения:
 - Google CDN,
 - Yandex CDN,
 - Microsoft CDN,
 - CDNJS CDN,
 - jQuery CDN,
 - Hosting.

Режим разработки

- **Включить режим разработки** – Если отметить, в подвале всех страниц сайта будет выводиться консоль, содержащая все уведомления сервера с замечаниями и PHP-ошибками. Режим разработки также отключает сжатие CSS и JS файлов. (Веб-мастеру и программисту. Не меняйте этот параметр, если не уверены в результате!)
- **Показывать ошибки только администратору.**
- **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 кэш, после внесения изменений в скрипты. (Веб-мастеру и программисту. Не меняйте этот параметр, если не уверены в результате!)
- **[Экстремальное кэширование](#)** – Внимание! Возможно ограничение функционала!

Используйте только после ознакомления с назначением данного параметра.

- **Включить сжатие HTML-контента** – Если отметить, сгенерированная HTML-страница будет сжиматься перед отправкой в веб-браузер клиента.
- **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!)

Информация о сайте

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++.

База данных

[{config}](#) – Настройки модулей

Файлы

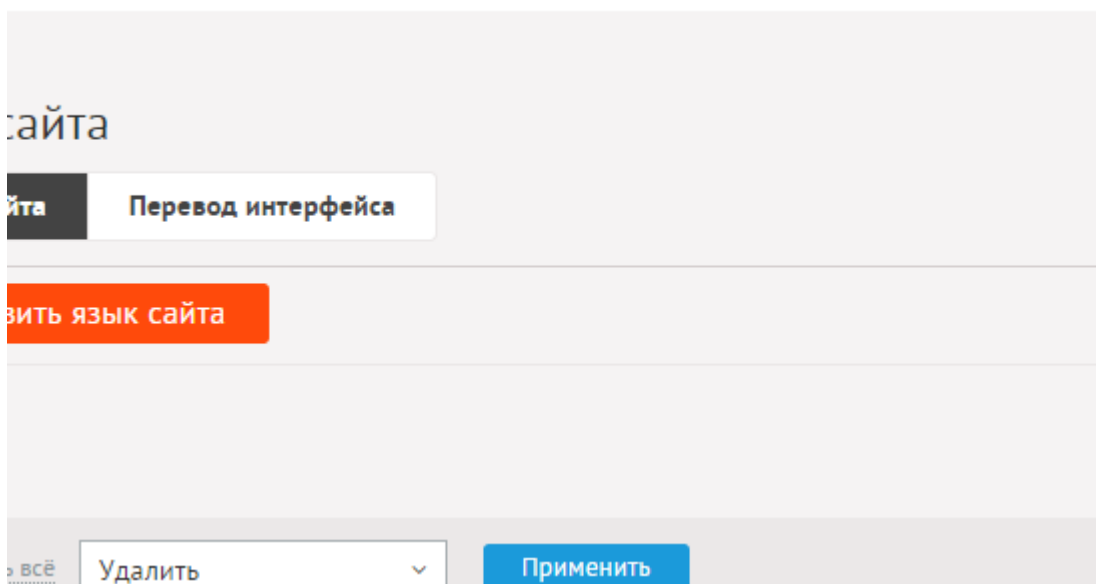
1. *config.php* – файл конфигурации;
2. *includes/config.php* – сохранение параметров сайта;
3. *modules/config/admin/config.admin.php* – редактирование параметров сайта;
4. *modules/config/admin/js/config.admin.js* – редактирование параметров сайта, JS-сценарий;
5. *modules/config/config.install.php*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

Языки сайта

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

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

Языки сайта



Языки сайта имеют следующие характеристики:

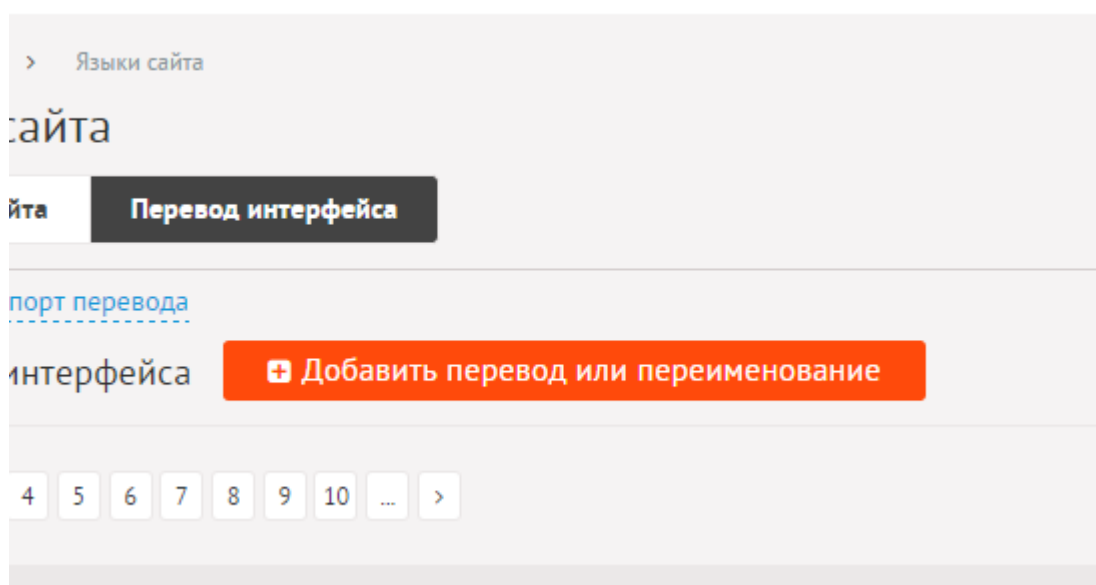
Основные:

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

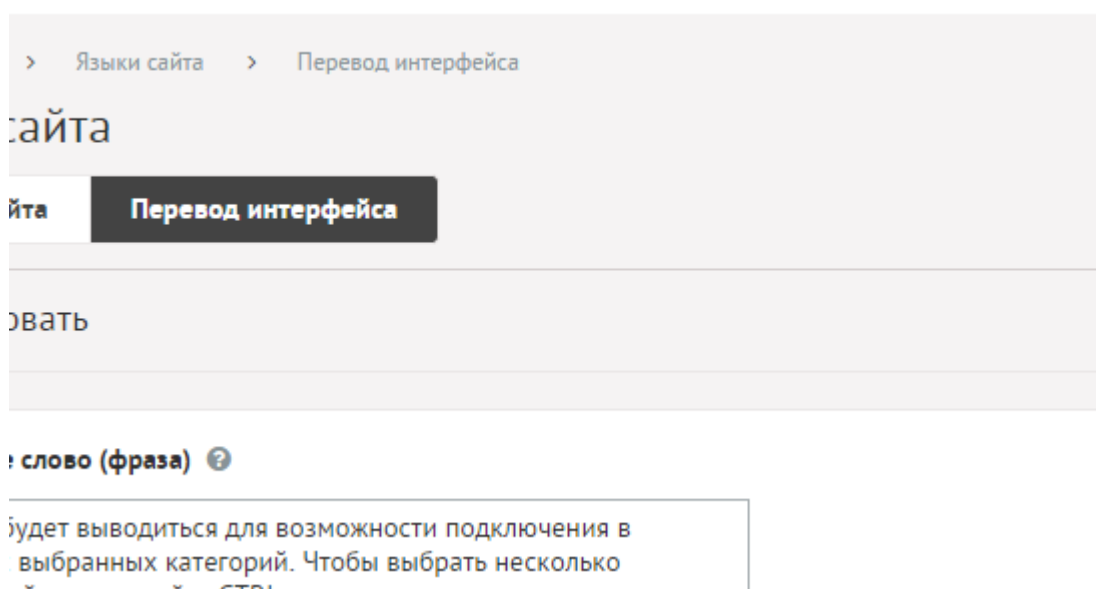
При добавлении и удалении языка сайта меняется структура базы данных, структура сайта. Нельзя удалить основной язык сайта.

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. Основной язык нельзя удалить.

Перевод интерфейса



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



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.

Свойства

var **is_ru = false** – это русская версия сайта.

Методы

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

- string \$name: текст для перевода
- string \$module_name: модуль
- boolean \$useradmin: выдавать форму для редактирования
- array \$args: аргументы

Для быстрого доступа к функции в основном объекте системы определена функция **_()**.

Пример:

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

string **base_admin ()** – Определяет язык версии административной панели для текущего пользователя.

void **import (string \$file_path, integer \$lang_id)** – Импортирует файл перевода.

- string \$file_path: путь до файла
- integer \$lang_id: ID языка, для которого загружается перевод

Пример:

```
// импортируем файл перевода langs/eng для языка интерфейса ID=2
$this->diafan->_languages->import(ABSOLUTE_PATH.'langs/eng', 2);
```

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:

```
// сокращенный вариант
$text = $this->diafan->_('Здравствуйте %s! У Вас %d новых сообщений.', false,
```

```
'Иван Иванович', 3);
```

Свойства

array **all** – массив данных обо всех языках, использующихся на сайте.

Пример:

```
print_r($this->diafan->_languages->all);
```

Результат:

```
Array
(
    [0] => Array
        (
            [id] => 1 // номер языка
            [name] => русский // название языка
            [shortname] => ru // сокращенное название, используется в URL
            [base_admin] => 1 // язык является основным для административной части
            [base_site] => 1 // язык является основным для сайта
            [page_act] => 1 // текущая страница существует для языкового зеркала
        )

    [1] => Array
        (
            [id] => 2
            [name] => английский
            [shortname] => eng
            [base_admin] => 0
            [base_site] => 0
            [page_act] => 1
        )
)
```

integer **site** – номер языка, использующегося как основной на сайте.

Пример:

```
if(_LANG == $this->diafan->_languages->site)
{
    echo 'Текущий язык является основным на сайте.';
}
```

integer **admin** – номер языка, использующегося как основной в административной части.

Пример:

```
if(_LANG == $this->diafan->_languages->admin)
{
    echo 'Текущий язык является основным для административной части.';
}
```

Шаблонные теги

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

show_block – выводит ссылки на альтернативные языковые версии сайта.

Атрибуты:

- **defer** – маркер отложенной загрузки шаблонного тега: **event** – загрузка контента только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** – загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка контента только по желанию пользователя;
- **defer_title** – текстовая строка, выводимая на месте появления загружаемого контента с помощью отложенной загрузки шаблонного тега;
- **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»

В шаблоне тега можно получить значение любого атрибута через переменную `$result["attributes"]["название атрибута"]`.

База данных

[{languages}](#) – Языки сайта

[{languages_translate}](#) – Перевод интерфейса

Файлы

1. *modules/languages/admin/js/languages.admin.translate.js* – редактирование списка языковых версий сайта, JS-сценарий;
2. *modules/languages/admin/languages.admin.php* – редактирование списка языковых версий сайта;

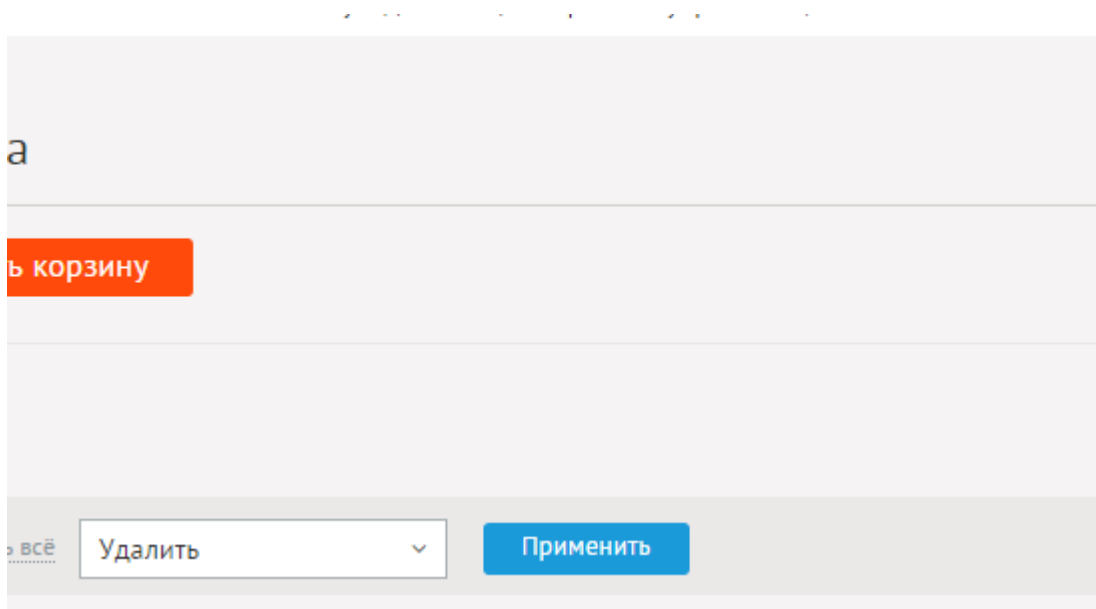
3. *modules/languages/admin/languages.admin.translate.php* – редактирование перевода интерфейса;
4. *modules/languages/languages.php* – контроллер;
5. *modules/languages/languages.export.php* – экспорт языкового файла;
6. *modules/languages/languages.inc.php* – подключение модуля;
7. *modules/languages/languages.install.php* – установка модуля;
8. *modules/languages/languages.model.php* – модель;
9. *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/js/trash.admin.js* – корзина с удаленными элементами, JS-сценарий;
2. *modules/trash/admin/trash.admin.php* – корзина с удаленными элементами;
3. *modules/trash/admin/trash.admin.count.php* – количество удаленных в корзину элементов;
4. *modules/trash/trash.install.php* – установка модуля.

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.

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.

Файловый менеджер

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

Новости

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

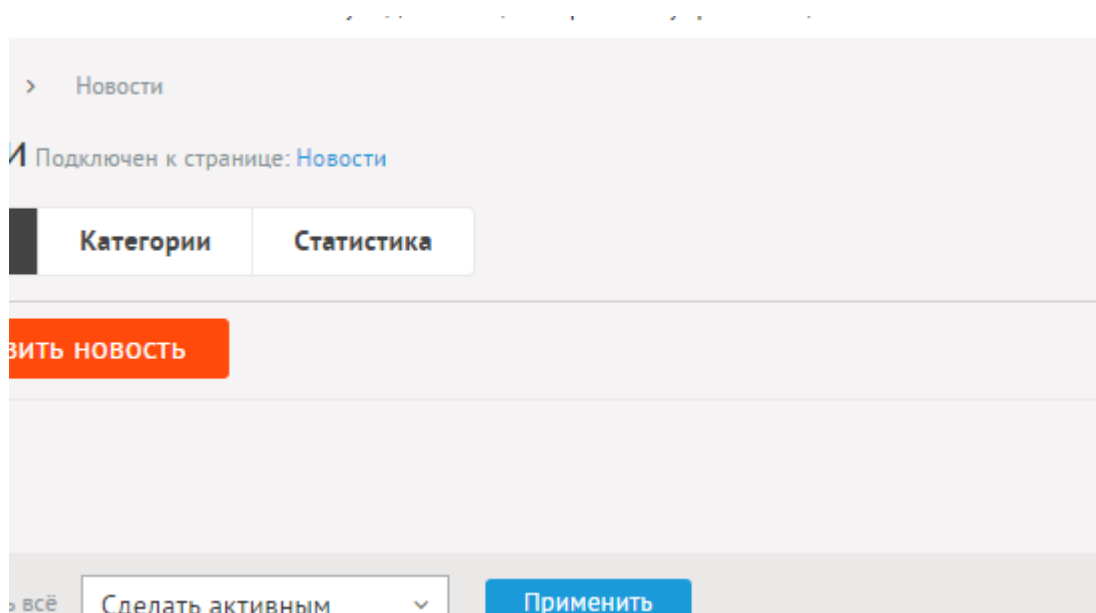
Модуль «Новости» отображает текстово-графические материалы с сортировкой по датам.

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

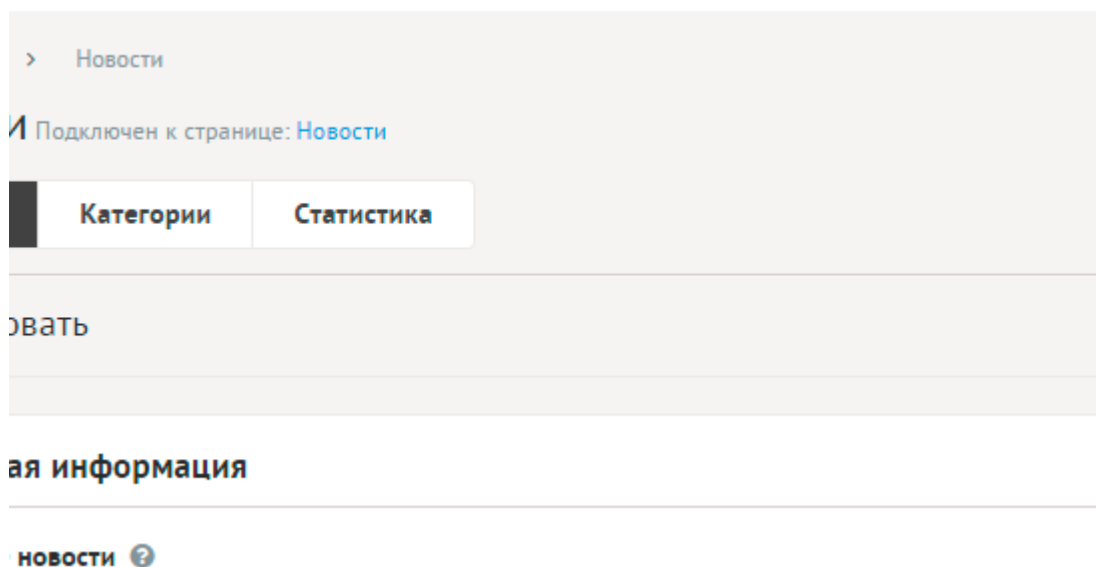
Модуль «Новости» можно прикреплять к различным страницам сайта. Если модуль прикреплен к нескольким страницам, в административной части модуля появляется специальный фильтр «Раздел сайта» с возможностью фильтрации вывода (списка новостей, категорий и настроек) по принадлежности к странице.

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

Новости



В списке новости сортируются по дате.



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.
- **Категория** – Категория, к которой относится новость. Список категорий редактируется во вкладке выше. 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».
- **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».
- **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»](#). Опция «HTML-код» позволяет отключить визуальный редактор для текущего поля. Значение этой настройки будет учитываться и при последующем редактировании. 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»](#). Опция «HTML-код» позволяет отключить визуальный редактор для текущего поля. Значение этой настройки будет учитываться и при последующем редактировании. Параметр имеет разные значения для разных языковых версий сайта.
- **Динамические блоки.**
- **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».
- **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.
- **Счетчик просмотров** – Количество просмотров на сайте текущей новости. 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 новостям».
- **Комментарии** – Комментарии, которые оставили пользователи к текущей новости. Параметр выводится, если в настройках модуля включен параметр «Показывать комментарии к новостям».

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*.
- **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.
- **Параметры SEO.**
- **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.

- **Канонический тег** – URL канонической страницы вида: <http://site.ru/pseudossylka/>, на которую переносится "ссылочный вес" данной страницы. Используется для страниц с похожим или дублирующимся контентом (SEO-специалисту). Параметр имеет разные значения для разных языковых версий сайта.
- **Псевдоссылка (ЧПУ)** – 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.
- **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).
- **Параметры показа.**
- **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).
- **Шаблоны.**
- **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!).
- **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*.

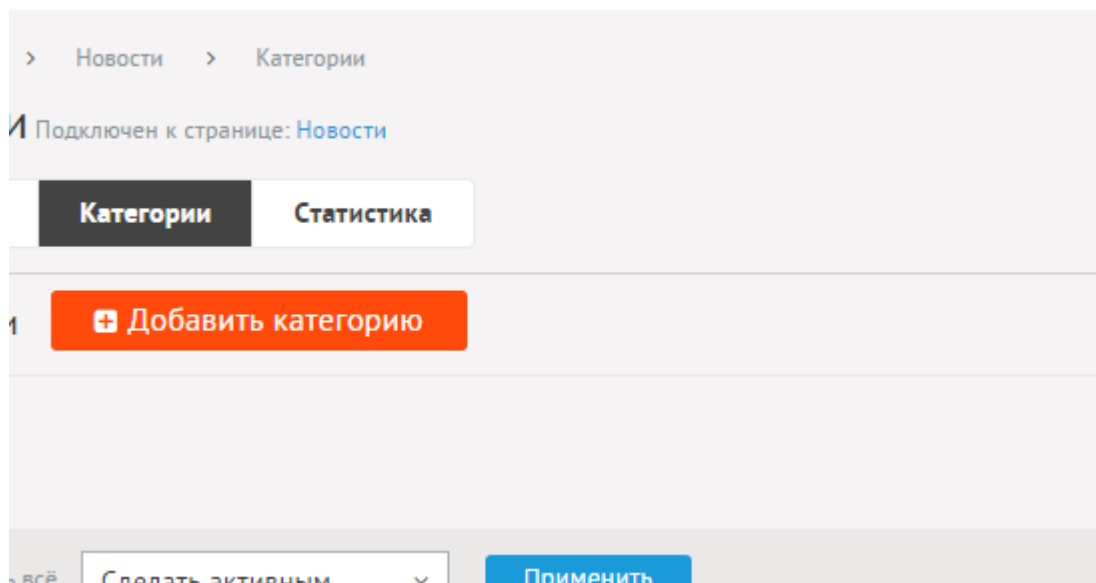
В модуль «Новости» встроена автоматическая публикация новостей в формате RSS.

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

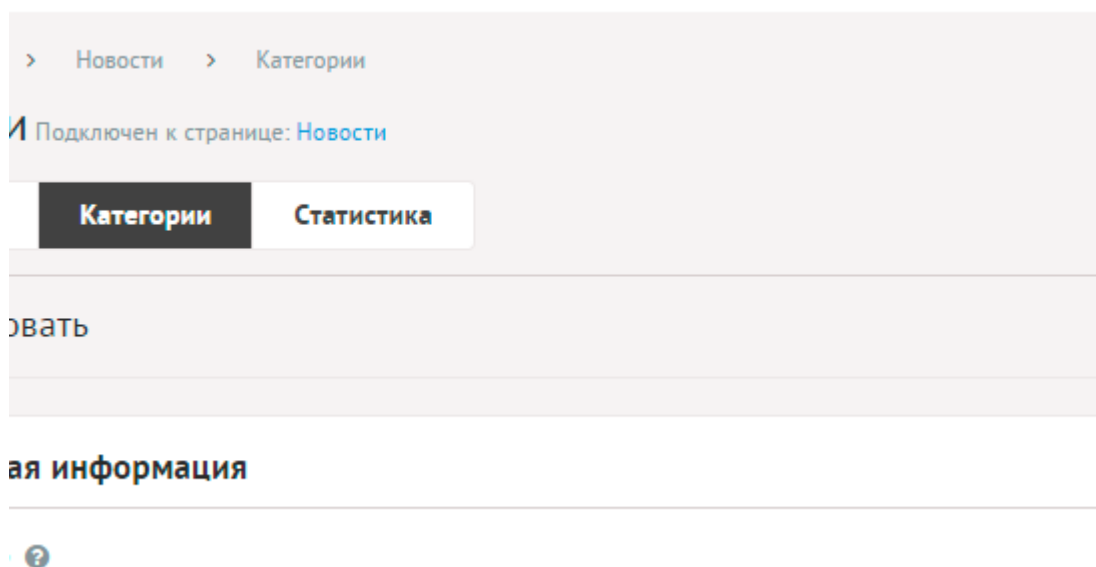
У англоязычной версии сайта адрес RSS-ленты новостей будет иметь вид:
<http://site.ru/eng/news/rss/>.

Категории

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



При удалении категории удаляются все вложенные подкатегории и новости.



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».
- **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»](#). Опция «HTML-код» позволяет отключить визуальный редактор для текущего поля. Значение этой настройки будет учитываться и при последующем редактировании. 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»](#). Опция «HTML-код» позволяет отключить визуальный редактор для текущего поля. Значение этой настройки будет учитываться и при последующем редактировании. Параметр имеет разные значения для разных языковых версий сайта.
- **Динамические блоки.**
- **Комментарии** – Выводится ссылка на комментарии к текущей категории или надпись «Комментариев нет» при их отсутствии. 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».
- **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*.
- **Раздел сайта** – Перенос категории на другую страницу сайта, к которой прикреплен модуль (администратору сайта).
- **Параметры SEO.**
- **Заголовок окна в браузере, тег Title** – Если не заполнен, тег *Title* будет автоматически сформирован как «Название категории – 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

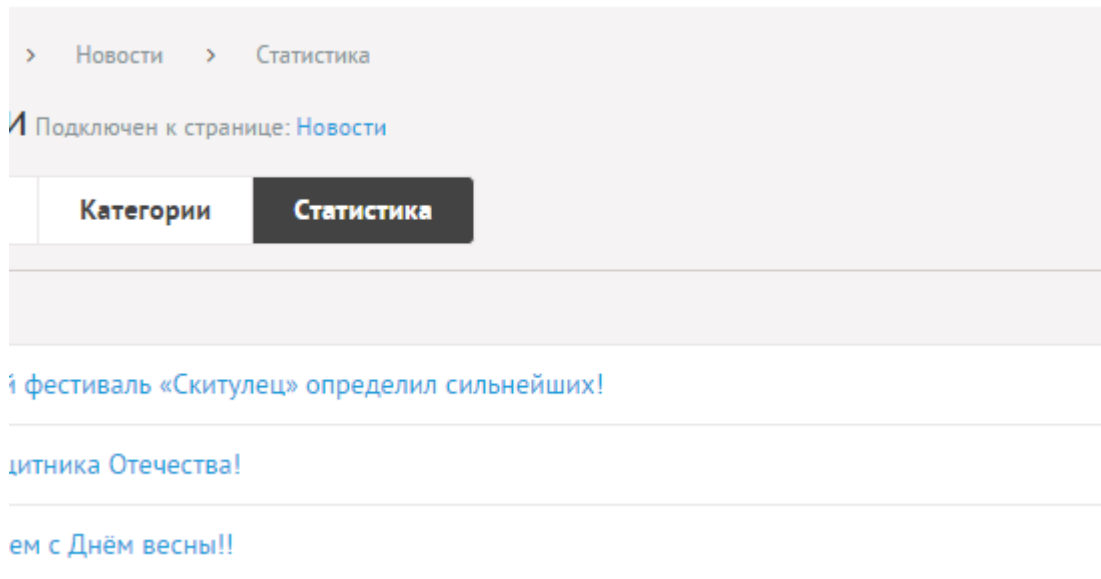
сформирован согласно шаблонам автоформирования из настроек модуля (SEO-специалисту). Параметр имеет разные значения для разных языковых версий сайта.

- **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.
- **Канонический тег** – URL канонической страницы вида: <http://site.ru/pseudossylka/>, на которую переносится "ссылочный вес" данной страницы. Используется для страниц с похожим или дублирующимся контентом (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.
- **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).
- **Параметры показа.**
- **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).
- **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).
- **Шаблоны.**
- **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 списка элементов** – Шаблон вывода контента модуля на странице элементов списка в категории (веб-мастеру и программисту, не меняйте этот параметр, если не

уверены в результате!). Значение параметра важно для AJAX.

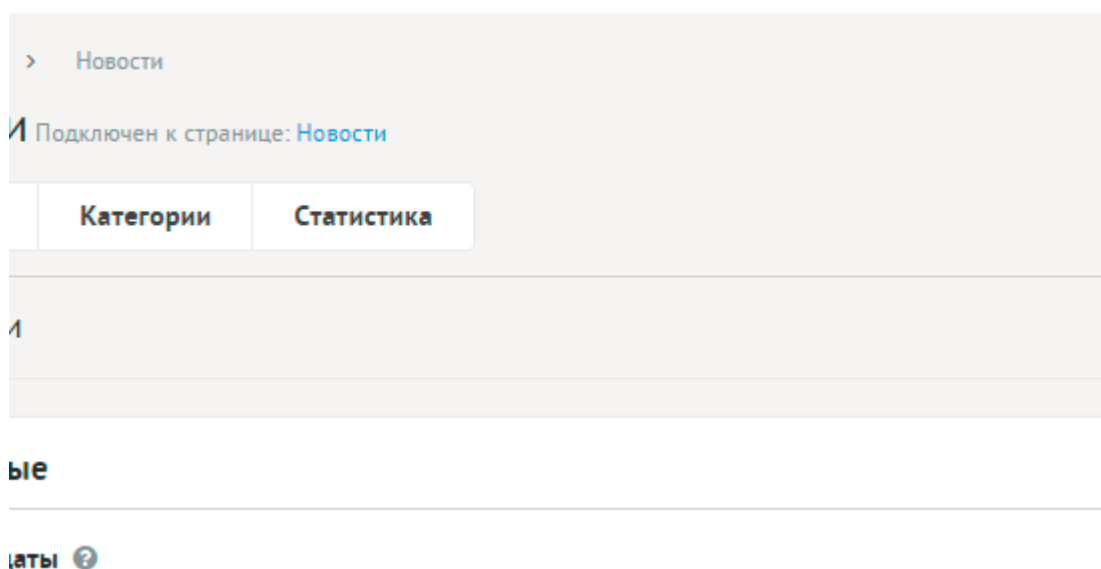
- **Шаблон страницы элемента** – Шаблон вывода контента модуля на странице отдельной новости, вложенной в текущую категорию (веб-мастеру и программисту, не меняйте этот параметр, если не уверены в результате!).

Статистика



Настройки

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



- **Основные.**
- **Формат даты** – Позволяет настроить отображение даты в модуле. Возможные значения:
 - 01.05.2016,
 - 01.05.2016 14:45,

- 1 мая 2016 г.,
- 1 мая,
- 1 мая 2016, понедельник,
- вчера 15:30,
- не отображать.
- **Количество новостей на странице** – Количество одновременно выводимых новостей в списке.
- **Включить «Показать ещё»** – На странице новостей появится кнопка «Показать ещё». Увеличивает количество одновременно выводимых новостей в списке.
- **Категории.**
- **Использовать категории** – Разделение новостей на категории, рубрики.
- **Количество категорий на странице** – Количество одновременно выводимых категорий в списке на первой страницы модуля. Параметр появляется, если отмечена опция «Использовать категории».
- **Количество новостей в списке категорий** – Количество новостей, выводимых в списке категорий на главной странице модуля. 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».
- **Количество новостей в списке вложенной категории** – Для первой страницы модуля и для страницы категории. Параметр появляется, если отмечена опция «Использовать категории».
- **Изображения.**
- **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).
- **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».
- **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».
- **Подключения.**
- **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 выставять 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.
- **Автогенерация для SEO.**
- **Шаблон для автоматического генерирования Title** – Если шаблон задан и для новости не прописан заголовок *Title*, то заголовок автоматически генерируется по шаблону. В шаблон можно добавить:

- %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 и для категории не прописан заголовок *Title*, то заголовок автоматически генерируется по шаблону. В шаблон можно добавить:

- %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 и для новости не заполнено поле *Keywords*, то поле *Keywords* автоматически генерируется по шаблону. В шаблон можно добавить:

- %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 и для категории не заполнено поле *Keywords*, то поле *Keywords* автоматически генерируется по шаблону. В шаблон можно добавить:

- %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 и для новости не заполнено поле *Description*, то поле *Description* автоматически генерируется по шаблону. В шаблон можно добавить:

- %name – название,
- %category – название категории,
- %parent_category – название категории верхнего уровня,
- %anons – краткое описание (SEO-специалисту).

Параметр имеет разные значения для разных языковых версий сайта.

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

- %name – название категории,
- %parent – название категории верхнего уровня,
- %anons – краткое описание (SEO-специалисту).

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

- **Оформление.**
- **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 списка элементов** – По умолчанию *modules/news/views/news.view.rows.php*. Параметр для разработчиков! Не устанавливайте, если не уверены в результате. Значение параметра важно для AJAX.
- **Шаблон для первой страницы модуля (если подключены категории)** – По умолчанию *modules/news/views/news.view.fitst_page.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.fitst_page.php*. Параметр для разработчиков! Не устанавливайте, если не уверены в результате. Значение параметра важно для AJAX.
- **Шаблон для страницы новости** – По умолчанию, *modules/news/views/news.view.id.php*. Параметр для разработчиков! Не устанавливайте, если не уверены в результате.
- **Дополнительно.**
- **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. Можно указать отрицательное значение, тогда будут исключены новости из указанного раздела. По умолчанию выбираются все страницы;
- **cat_id** – категории новостей, если в настройках модуля отмечено «Использовать категории». Идентификаторы категорий перечисляются через запятую. Можно указать отрицательное значение, тогда будут исключены новости из указанной категории. Можно указать значение **current**, тогда будут показаны новости из текущей (открытой) категории или из всех категорий, если ни одна категория не открыта. По умолчанию категория не учитывается, выводятся все новости;
- **sort** – сортировка новостей: **date** – по дате (по умолчанию), **keywords** – новости,

похожие по названию для текущей страницы (должен быть подключен модуль «Поиск по сайту» и проиндексированы новости);

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

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

— **only_module** – выводить блок только на странице, к которой прикреплен модуль «Новости»: **true** – выводить блок только на странице модуля, по умолчанию блок будет выводиться на всех страницах;

— **tag** – тег, прикрепленный к новостям;

— **defer** – маркер отложенной загрузки шаблонного тега: **event** – загрузка контента только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** – загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка контента только по желанию пользователя;

— **defer_title** – текстовая строка, выводимая на месте появления загружаемого контента с помощью отложенной загрузки шаблонного тега;

— **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

В шаблоне тега можно получить значение любого атрибута через переменную `$result["attributes"]["название атрибута"]`.

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;
- **defer** – маркер отложенной загрузки шаблонного тега: **event** – загрузка контента только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** – загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка контента только по желанию пользователя;
- **defer_title** – текстовая строка, выводимая на месте появления загружаемого контента с помощью отложенной загрузки шаблонного тега;
- **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

В шаблоне тега можно получить значение любого атрибута через переменную `$result["attributes"]["название атрибута"]`.

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»;
- **detail** – детализация (**day** – дни, **month** – месяца (по умолчанию), **year** – годы);
- **only_module** – выводить блок только на странице «Новости»: **true** – выводить блок только на странице модуля, по умолчанию блок будет выводиться на всех страницах;
- **defer** – маркер отложенной загрузки шаблонного тега: **event** – загрузка контента

только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** – загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка контента только по желанию пользователя;

— **defer_title** – текстовая строка, выводимая на месте появления загружаемого контента с помощью отложенной загрузки шаблонного тега;

— **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

В шаблоне тега можно получить значение любого атрибута через переменную `$result["attributes"]["название атрибута"]`.

База данных

[{news}](#) – Новости

[{news_category}](#) – Категории новостей

[{news_category_parents}](#) – Родительские связи категорий новостей

[{news_category_rel}](#) – Связи новостей и категорий

[{news_counter}](#) – Счетчик просмотров новостей

[{news_rel}](#) – Связи похожих 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/js/news.show_calendar_day.js* – JS-сценарий модуля;

7. *modules/news/news.php* – контроллер;

8. *modules/news/news.action.php* – обработка Ajax-запросов;

9. *modules/news/news.install.php* – установка 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.rows.php* – шаблон элементов в списке новостей;

19. *modules/news/views/news.view.rows_block.php* – шаблон блока новостей;

20. *modules/news/views/news.view.rows_block_rel.php* – шаблон блока похожих новостей;

21. *modules/news/views/news.view.show_block.php* – шаблон блока новостей;

22. *modules/news/views/news.view.show_block_rel.php* – шаблон блока похожих новостей;

23. *modules/news/views/news.view.show_calendar.php* – шаблон календаря архива новостей;

24. *modules/news/views/news.view.show_calendar_day.php* – шаблон календаря архива новостей по дням.

Магазин

- [Товары](#)
- [Категории](#)
- [Производители](#)
- [Характеристики](#)
- [Подключение](#)
- [Заказы](#)
- [Форма оформления заказа](#)
- [Отчеты](#)
- [Список пожеланий](#)
- [Список ожиданий](#)
- [Статусы заказа](#)
- [Скидки](#)
- [Валюты](#)
- [Способы доставки](#)
- [Сопутствующие услуги](#)
- [Импорт/экспорт](#)
- [Экспорт товаров для Яндекс.Маркет](#)
- [Интеграция с системой 1С:Предприятие](#)
- [Интеграция с системой МойСклад](#)
- [Настройки модуля](#)
- [Шаблонные теги](#)
- [База данных](#)
- [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

Товары

Интернет-магазин

Подключен к странице: Интернет-магазин

СкидкиПроизводителиКатегорииХарактеристикиИмпорт

Добавить товар

Сделать активным

Применить

Товары

Интернет-магазин

Подключен к странице: Интернет-магазин

СкидкиПроизводителиКатегорииХарактеристикиИмпорт

Добавить

Дополнительная информация

Товары имеют следующие характеристики.

Основные:

- **Название** – Используется в заголовках, ссылках на товар, при заказе. Параметр имеет разные значения для разных языковых версий сайта.
- **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 получат уведомления о поступлении товара на указанный электронный ящик.
- **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.
- **Единица измерения** – Параметр имеет разные значения для разных языковых версий сайта.
- **Категория** – Категория, к которой относится товар. Список категорий редактируется во вкладке выше. 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».
- **Производитель** – Список производителей редактируется во вкладке выше.
- **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.
- **Вес** – Используется для расчета стоимости доставки.
- **Длина** – Используется для расчета стоимости доставки.
- **Ширина** – Используется для расчета стоимости доставки.
- **Высота** – Используется для расчета стоимости доставки.
- **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».
- **Общие скидки** – Список глобальных скидок. Возможность подключить или отключить скидку для товара.
- **Сопутствующие услуги** – Возможность добавить услуги к товару за дополнительную стоимость
- **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.

- **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»](#). Опция «HTML-код» позволяет отключить визуальный редактор для текущего поля. Значение этой настройки будет учитываться и при последующем редактировании. 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»](#). Опция «HTML-код» позволяет отключить визуальный редактор для текущего поля. Значение этой настройки будет учитываться и при последующем редактировании. Параметр имеет разные значения для разных языковых версий сайта.
- **Динамические блоки.**
- **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».
- **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.
- **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».
- **Выгружать в Google Merchant** – Параметр разрешит или запретит выгружать этот товар. Параметр выводится, если в настройках модуля отмечена опция «Подключить Google Merchant» и параметр «Выгружать товары в Google Merchant» определен как «только помеченные».
- **Поля для Google Merchant** – Параметры, необходимые для формирования информации для системы Google Merchant. Параметр выводится, если в настройках модуля отмечена опция «Подключить Google Merchant». Параметр появляется, если отмечена опция «Выгружать в Google Merchant».

Дополнительные:

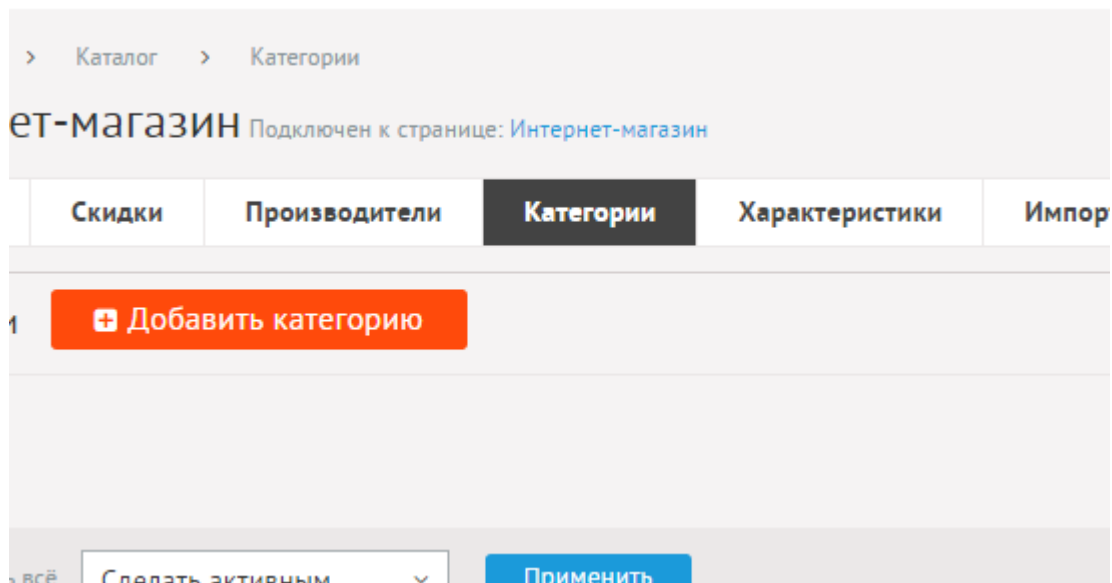
- **Номер** – Номер элемента в БД (веб-мастеру и программисту).
- **Редактор** – Изменяется после первого сохранения. Показывает, кто из администраторов сайта первый правил текущую страницу.
- **Время последнего изменения** – Изменяется после сохранения элемента. Отдается в заголовке *Last Modify*.
- **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.
- **Параметры SEO.**
- **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.
- **Канонический тег** – URL канонической страницы вида: <http://site.ru/pseudossylka/>, на которую переносится "ссылочный вес" данной страницы. Используется для страниц с похожим или дублирующимся контентом (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.
- **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).
- **Параметры показа.**
- **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).

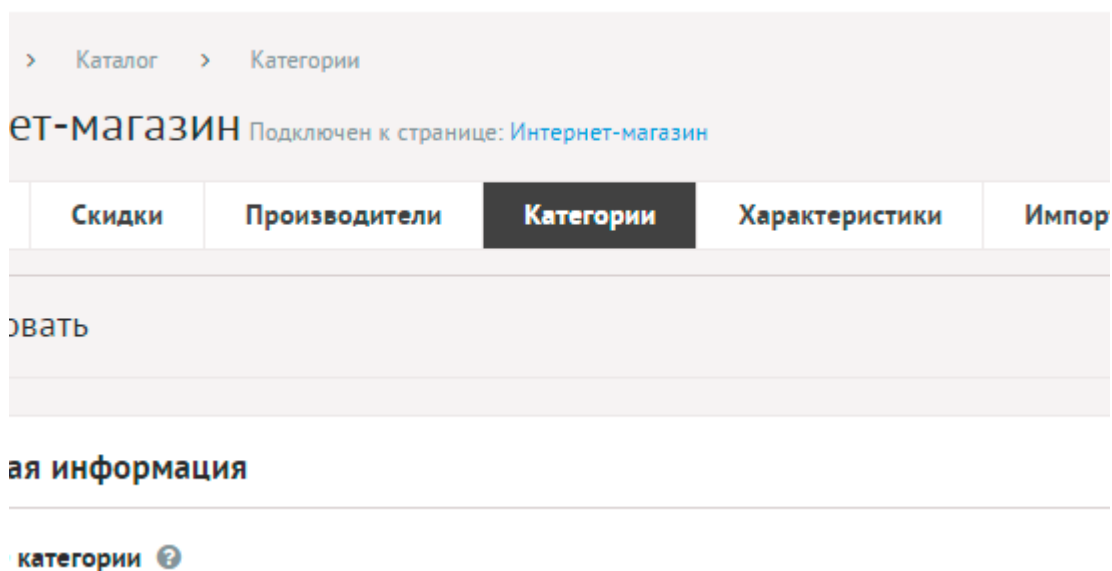
- **Не показывать на карте сайта** – Скрывает отображение ссылки на товар в файле *sitemap.xml* и [модуле «Карта сайта»](#).
- **Выгружать в Яндекс.Маркет** – Параметр разрешит или запретит выгружать этот товар. Параметр выводится, если в настройках модуля отмечена опция «Подключить Яндекс Маркет» и 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).
- **Оформление.**
- **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!).
- **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*.

Категории

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



При удалении категории удаляются все вложенные подкатегории и товары.



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».
- **К категории прикреплены характеристики товаров** – Список характеристик, применимых к товарам текущей категории.

- **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»](#). Опция «HTML-код» позволяет отключить визуальный редактор для текущего поля. Значение этой настройки будет учитываться и при последующем редактировании. 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»](#). Опция «HTML-код» позволяет отключить визуальный редактор для текущего поля. Значение этой настройки будет учитываться и при последующем редактировании. Параметр имеет разные значения для разных языковых версий сайта.
- **Динамические блоки.**
- **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».
- **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*.
- **Раздел сайта** – Перенос категории на другую страницу сайта, к которой прикреплен модуль (администратору сайта).
- **Параметры SEO.**
- **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.

- **Канонический тег** – URL канонической страницы вида: <http://site.ru/pseudossylka/>, на которую переносится "ссылочный вес" данной страницы. Используется для страниц с похожим или дублирующимся контентом (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.
- **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).
- **Параметры показа.**
- **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».
- **Выгружать в Google Merchant** – Параметр разрешит или запретит выгружать эту категорию. Параметр выводится, если в настройках модуля отмечена опция «Подключить Google Merchant» и параметр «Выгружать категории в Google Merchant» определен как «только помеченные».
- **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).
- **Оформление.**
- **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 списка элементов** – Шаблон вывода контента модуля на странице элементов списка в категории (веб-мастеру и программисту, не меняйте этот параметр, если не уверены в результате!). Значение параметра важно для AJAX.
- **Шаблон страницы элемента** – 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!).

Производители

Товару можно задать производителя. Ссылка на все товары производителя появиться у каждого товара.

Производители имеют следующие характеристики.

Основные:

- **Название** – Используется в заголовках, ссылках на производителя. Параметр имеет разные значения для разных языковых версий сайта.
- **Опубликовать на сайте** – Если не отмечена, производителя 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 текущего производителя.
- **Изображения** – Возможность загрузки нескольких изображений. Варианты размера загружаемых изображений определяются в настройках. Параметр выводится, если в настройках модуля отмечена опция «Использовать изображения для производителей».
- **Динамические блоки.**
- **Категории** – Категории, в которых используется производитель. Чтобы выбрать несколько категорий, удерживайте CTRL.
- **Полное описание** – Если отметить «Применить типограф», контент будет отформатирован согласно правилам экранной типографики с помощью [веб-сервиса «Типограф»](#). Опция «HTML-код» позволяет отключить визуальный редактор для текущего поля. Значение этой настройки будет учитываться и при последующем редактировании. Параметр имеет разные значения для разных языковых версий сайта.
- **Индексирование для поиска** – Производитель автоматически индексируется для модуля «Поиск по сайту» при внесении изменений.
- **Индексирование для карты сайта** – Производитель 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

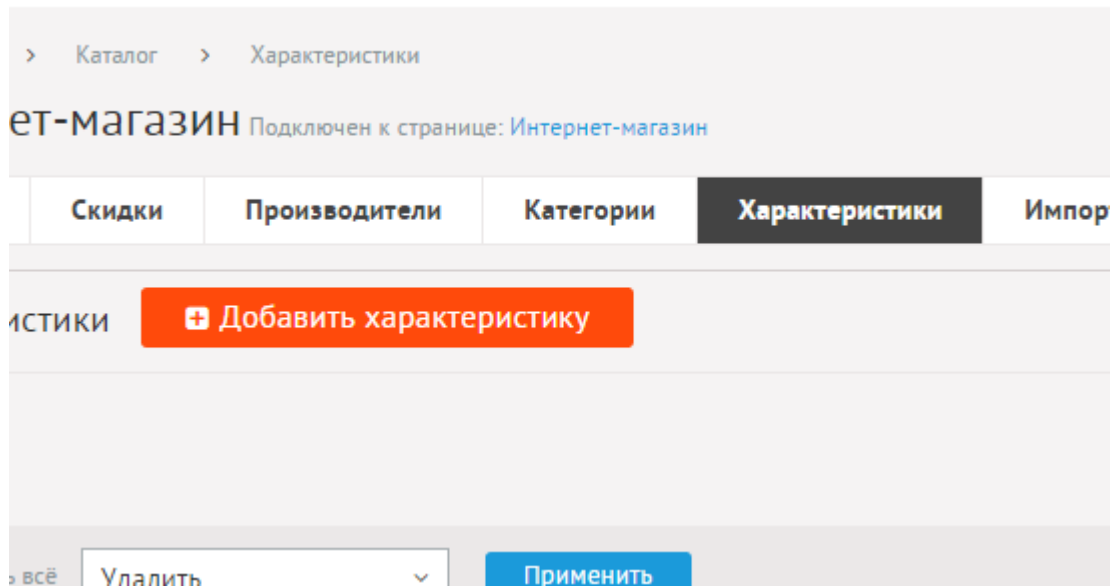
администраторов сайта первый правил текущую страницу.

- **Время последнего изменения** – Изменяется после сохранения элемента. Отдается в заголовке *Last Modify*.
- **Параметры SEO.**
- **Заголовок окна в браузере, тег Title** – Если не заполнен, тег *Title* будет автоматически сформирован как «Название бренда – 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.
- **Канонический тег** – URL канонической страницы вида: *http://site.ru/pseudossylka/*, на которую переносится "ссылочный вес" данной страницы. Используется для страниц с похожим или дублирующимся контентом (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.
- **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).
- **Параметры показа.**
- **Не показывать на карте сайта** – Скрывает отображение ссылки на производителя в файле *sitemap.xml* и [модуле «Карта сайта»](#).
- **Раздел сайта** – Перенос производителя на другую страницу сайта, к которой прикреплен модуль (администратору сайта).
- **Сортировка: установить перед** – Редактирование порядка следования производителя в списке. Поле доступно для редактирования только для производителей, отображаемых на сайте.
- **Идентификатор для импорта** – Можно заполнить для идентификации производителя при импорте (администратору сайта).
- **Шаблоны.**
- **Шаблон страницы** – Возможность подключить для страницы производителя 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!).

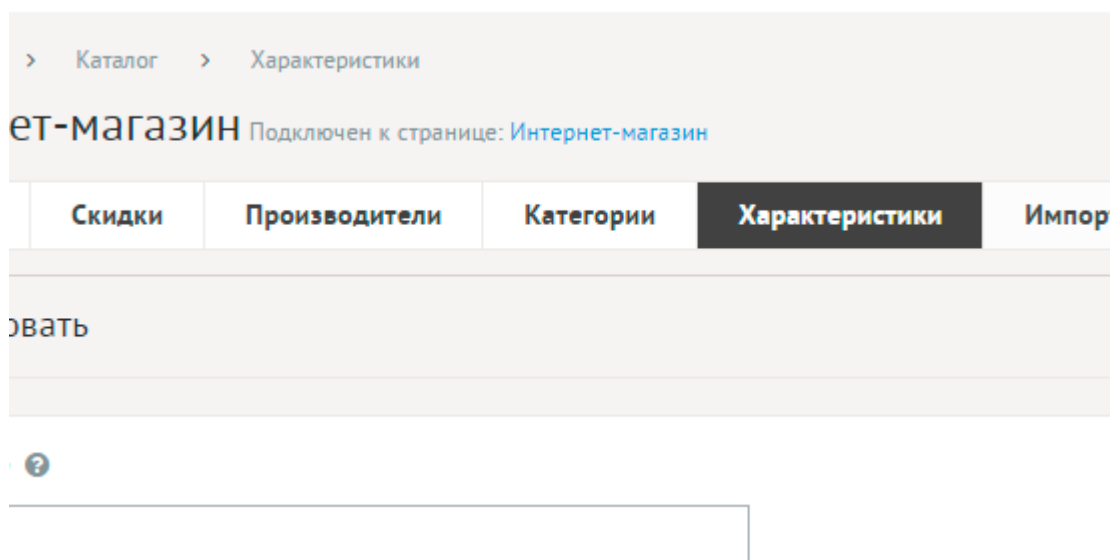
- **Шаблон модуля** – Шаблон вывода контента модуля на странице списка товаров производителя (веб-мастеру и программисту, не меняйте этот параметр, если не уверены в результате!).

Характеристики



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.

Чтобы прикрепить несколько характеристик к одной категории или, наоборот, открепить от категории, можно воспользоваться групповой операцией.



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,
 - электронный ящик,
 - телефон,
 - 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.
- **Раздел сайта** – Раздел сайта, к которому применяется характеристика.
- **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.
- **Показывать на странице товара** – Если отмечено, данная характеристика будет отображаться на странице товара.
- **Отображать параметры в блоке для сортировки товаров** – Позволяет выводить характеристику в виде ссылки для сортировки товаров по значению характеристики.
- **Выгружать в файле YML (Яндекс.Маркет)** – Характеристика будет выгружена в [элемент param](#).
- **Название для файла YML (Яндекс.Маркет).**
- **Единица измерения для файла YML (Яндекс.Маркет).**
- **Описание характеристики** – 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.

Подключение

Подключаемая часть – файл *modules/shop/shop.inc.php*. В нем описан класс *Shop_inc*. В модуле к объекту класса можно обратиться через переменную `$this->diafan->_shop`. Экземпляр класса создается при первом вызове переменной.

Методы подключения интернет-магазина делаются на две части по смысловой нагрузке:

- методы для работы с ценами (*modules/shop/inc/shop.inc.price.php*);
- методы для работы с заказами (*modules/shop/inc/shop.inc.order.php*).

Кроме того модуль «Интернет-магазин» включает два модуля, которые тоже имеют свои подключения:

- корзина *modules/cart/cart.inc.php*;
- список пожеланий *modules/wishlist/wishlist.inc.php*.

Работа с ценами

К методам для работы с ценами нужно добавлять префикс `price_`. Например, `$this->diafan->_shop->price_get()`.

Методы

array **get (integer \$good_id, array \$params, [boolean \$current_user = true])** – Получает цену товара с указанными параметрами для пользователя.

- integer \$good_id: номер товара
- array \$params: параметры, влияющие на цену
- boolean \$current_user: текущий пользователь

Пример:

```
// получаем цену товара ID=3 цвета (ID=6) синего (ID=15),
// размера (ID=5) XS (ID=16). При выборе цены принимаются во внимание
// персональные скидки для текущего пользователя
$price = $this->diafan->_shop->price_get(3, array(6 => 15, 5 => 16));

print_r($price);
/* выведет:
Array
(
    [id] => 39
    [price_id] => 39
    [count_goods] => 5
    [price] => 1390
    [old_price] => 1500
    [discount_id] => 1
) */
```

array **get_person_discounts ()** – Возвращает идентификаторы персональных скидок, применимые для текущего пользователя.

Пример:

```
// получаем идентификаторы персональных скидок
$person_discount_ids = $this->diafan->_shop->price_get_person_discounts();

$cache_meta = array(
    "name" => "list",
    // ...
    "discounts" => $person_discount_ids
);
// кеширование
if (! $result = $this->diafan->_cache->get($cache_meta, "shop"))
{
    // ...
    $this->diafan->_cache->save($result, $cache_meta, "shop");
}
```

```
}
```

array **get_all (integer \$good_id, [integer \$current_user = true])** – Получает все цены товара для пользователя.

- integer \$good_id: номер товара
- integer \$current_user: пользователь, для которого определяется цена

Пример:

```
// получаем все цены для товара ID=12, при выборе цены принимаются во внимание
// персональные скидки для текущего пользователя
$prices = $this->diafan->_shop->price_get_all(12);
```

```
print_r($prices);
```

```
/* выведет:
```

```
Array
```

```
(
```

```
  [0] => Array
```

```
  (
```

```
    [id] => 94
```

```
    [good_id] => 12
```

```
    [price] => 5490
```

```
    [old_price] => 5990
```

```
    [count_goods] => 0
```

```
    [price_id] => 12
```

```
    [date_start] => 0
```

```
    [date_finish] => 0
```

```
    [discount] => 0
```

```
    [discount_id] => 4
```

```
    [person] => 0
```

```
    [role_id] => 0
```

```
    [currency_id] => 0
```

```
    [import_id] =>
```

```
    [trash] => 0
```

```
  )
```

```
  [1] => Array
```

```
  (
```

```
    [id] => 95
```

```
    [good_id] => 12
```

```
    [price] => 5490
```

```
    [old_price] => 5990
```

```
    [count_goods] => 0
```

```
    [price_id] => 13
```

```
    [date_start] => 0
```

```
    [date_finish] => 0
```

```
    [discount] => 0
```

```
    [discount_id] => 4
```

```
    [person] => 0
```

```

        [role_id] => 0
        [currency_id] => 0
        [import_id] =>
        [trash] => 0
    )
)*/

```

void **prepare_all (integer \$good_id)** – Подготавливает все цены товара для пользователя.

- integer \$good_id: номер товара

Пример:

```

// в этом примере будет выполнено три SQL-запроса к базе данных для получения цен
для всех заданных товаров
$ids = array(3, 5, 7);
foreach($ids as $id)
{
    $prices[$id] = $this->diafan->_shop->price_get_all($id);
}

```

Пример:

```

// в этом примере будет выполнен один SQL-запрос к базе данных для получения цен
для всех заданных товаров
$ids = array(3, 5, 7);
foreach($ids as $id)
{
    $this->diafan->_shop->price_prepare_all($id);
}
foreach($ids as $id)
{
    $prices[$id] = $this->diafan->_shop->price_get_all($id);
}

```

array **get_base (integer \$good_id, [boolean \$base_currency = false])** – Получает основы для цен на товар (указываемые в панели администрирования).

- integer \$good_id: номер товара
- boolean \$base_currency: показывать результаты в основной валюте

Пример:

```

// получаем все цены для товара ID=12 без учета скидок (базовые цены)
$prices = $this->diafan->_shop->price_get_base(12);

print_r($prices);
/* выведет:
Array

```

```
(
  [0] => Array
  (
    [id] => 12
    [price_id] => 12
    [price] => 5990
    [currency_id] => 0
    [count_goods] => 0
    [good_id] => 12
    [currency_name] => руб.
    [param] => Array
      (
        [2] => 2
      )
  )

  [1] => Array
  (
    [id] => 13
    [price_id] => 13
    [price] => 5990
    [currency_id] => 0
    [count_goods] => 0
    [good_id] => 12
    [currency_name] => руб.
    [param] => Array
      (
        [2] => 1
      )
  )

) */
```

array **prepare_base(integer \$good_id)** – Подготавливает основы для цен на товар (указываемые в панели администрирования).

- integer \$good_id: номер товара

Пример:

```
// в этом примере будет выполнено три SQL-запроса к базе данных для получения
базовых цен для всех заданных товаров
$ids = array(3, 5, 7);
foreach($ids as $id)
{
  $prices[$id] = $this->diafan->_shop->price_get_base($id);
}
```

Пример:

```
// в этом примере будет выполнен один SQL-запрос к базе данных для получения базовых цен для всех заданных товаров
```

```
$ids = array(3, 5, 7);  
foreach($ids as $id)  
{  
    $this->diafan->_shop->price_prepare_base($id);  
}  
foreach($ids as $id)  
{  
    $prices[$id] = $this->diafan->_shop->price_get_base($id);  
}
```

void **calc** ([integer \$good_id = 0], [integer \$discount_id = 0], [integer \$currency_id = 0]) –
Рассчитывает все возможные вариации цен и записывает их в базу данных.

- integer \$good_id: номер товара, если не задан, цены рассчитываются для всех товаров
- integer \$discount_id: номер скидки
- integer \$currency_id: номер валюты, если нужно изменить цены, указанные в валюте

Пример:

```
// после сохранения изменений для скидки ID=5  
// рассчитываем цены для всех товаров с учетом этой скидки  
$this->diafan->_shop->price_calc(0, 5);
```

integer **insert** (integer \$good_id, float \$price, float \$old_price, integer \$count, [integer \$params = array()], [integer \$currency_id = 0], [integer \$import_id = ''], [integer \$image_id = 0]) –
Добавляет базовую цену для товара.

- integer \$good_id: номер товара
- float \$price: цена
- float \$old_price: старая цена
- integer \$count: количество товара
- integer \$params: дополнительные характеристики, учитываемые в цене
- integer \$currency_id: номер валюты
- integer \$import_id: ID цены для импорта
- integer \$image_id: ID изображения, прикрепляемого к цене

Пример:

```
// записываем цену 1500 руб. для товара ID=13, количество товара 5 шт.  
// цвет (ID=6) синий (ID=15), размер (ID=5) XS (ID=16)  
$price_id = $this->diafan->_shop->price_insert(13, 1500, 5, array(6 => 15, 5 => 16));
```

void **send_mail_waitlist** (integer \$good_id, array \$params, [array \$row = array()]) – Отправляет уведомления о поступлении товара.

- integer \$good_id: идентификатор товара
- array \$params: дополнительные характеристики, влияющие на цену
- array \$row: данные о товаре

mixed (string|float) **format (float \$price, [boolean \$float = false])** – Форматирует цену согласно настройкам модуля.

- float \$price: цена
- boolean \$float: возвращаемый результат: **true** - дискретное число, по умолчанию - строка

Пример:

```
echo $this->diafan->_shop->price_insert(23000.5);  
// выведет: 23 000,50
```

integer **num_decimal_places ([float \$price = false])** – Возвращает количество знаков в цене после запятой, установленное в настройках модуля.

- float \$price: цена

Работа с заказами

К методам для работы с заказами нужно добавлять префикс `order_`. Например, `$this->diafan->_shop->order_pay()`.

Методы

array **get (integer \$order_id)** – Получает все данные о товарах, дополнительных услугах, доставке и скидках в заказе.

- integer \$order_id: номер заказа

array **get_param (integer \$order_id)** – Получает все данные из формы оформления заказа.

- integer \$order_id: номер заказа

void **pay (integer \$order_id)** – Оплата заказ (смена статуса на «В обработке»).

- integer \$order_id: номер заказа

Пример:

```
// оплата заказа №12 (смена статуса, уменьшение количества товара на складе)  
$this->diafan->_shop->order_pay(12);
```

void **set_status (array \$order, array \$status)** – Установка статуса.

- array \$order: информация о заказе
- array \$status: информация о статусе заказа

array **details (integer \$order_id)** – Возврат информации о плательщике.

- integer \$order_id: ID заказа

Корзина

К объекту класса можно обратиться через переменную `$this->diafan->_cart`. Экземпляр класса создается при первом вызове переменной.

Методы

mixed get ([integer \$id = 0], [mixed \$param = false], [mixed \$additional_cost = false], [string \$name_info = "]) – Возвращает информацию из корзины.

- integer \$id: номер товара
- mixed \$param: характеристики товара, учитываемые в заказе
- mixed \$additional_cost: сопутствующие услуги
- string \$name_info: тип информации (count - количество, is_file - это товар-файл)

Пример:

```
// запрашиваем все товары, находящиеся в корзине
$cart = $this->diafan->_cart->get();
print_r($cart);
/* выведет:
Array
(
    [38] => Array
        (
            [a:0:{}] => Array
                (
                    [price_id] => 39
                    [count] => 1
                    [is_file] => 0
                )
        )

    [49] => Array
        (
            [a:0:{}] => Array
                (
                    [price_id] => 60
                    [count] => 1
                    [is_file] => 0
                )
        )
)*/

// запрашиваем количество товара ID=38 в корзине
echo $this->diafan->_cart->get(38, array(), "count");
// выведет: 1
```

integer get_count () – Возвращает количество товаров в корзине.

Пример:

```
echo 'В корзине '.$this->diafan->_cart->get_count().' товара';  
// выведет: В корзине 2 товара
```

float **get_summ ()** – Возвращает общую стоимость товаров в корзине.

Пример:

```
echo 'В корзине товаров на сумму '.$this->diafan->_cart->get_summ().' руб.';  
// выведет: В корзине товаров на сумму 2 738 руб.
```

void **set ([mixed \$value = array()], [integer \$id = 0], [mixed \$param = false], [mixed \$additional_cost = false], [string \$name_info = "])** – Записывает данные в корзину.

- mixed \$value: данные
- integer \$id: номер товара
- mixed \$param: характеристики товара, учитываемые в заказе
- mixed \$additional_cost: сопутствующие услуги
- string \$name_info: тип информации (count - количество, is_file - это товар-файл)

Пример:

```
// обновляем количество товара ID=38 в корзине  
// или добавляем его туда, если товара в корзине нет  
$this->diafan->_cart->set(3, 38, array(), "count");  
if($err = $this->diafan->_cart->set($cart, 38, array()))  
{  
    echo 'Ошибка: '.$err;  
}  
  
// обновляем количество товара и пометку, что товар является файлом  
// или добавляем его туда  
$cart = array(  
    "count" => 3,  
    "is_file" => true,  
);  
if($err = $this->diafan->_cart->set($cart, 38, array()))  
{  
    echo 'Ошибка: '.$err;  
}  
  
// удаляем товар ID=38 из корзины  
$this->diafan->_cart->set(0, 38, array(), "count");  
  
// очищаем корзину  
$this->diafan->_cart->set();
```

void **write ()** – Записывает информацию о корзине в хранилище.

Пример:

```
// очищаем корзину
$this->diafan->_cart->set();

// записываем данные, установленные функцией set()
$this->diafan->_cart->write();
```

Список пожеланий

К объекту класса можно обратиться через переменную `$this->diafan->_wishlist`. Экземпляр класса создается при первом вызове переменной.

Методы

mixed `get ([integer $id = 0], [mixed $param = false], [mixed $additional_cost = false], [string $name_info = "])` – Возвращает информацию из списка пожеланий.

- integer `$id`: номер товара
- mixed `$param`: характеристики товара, учитываемые в заказе
- mixed `$additional_cost`: сопутствующие услуги
- string `$name_info`: тип информации (`count` - количество, `is_file` - это товар-файл)

Пример:

```
// запрашиваем все товары, находящиеся в списке пожеланий
$wishlist = $this->diafan->_wishlist->get();
print_r($wishlist);
/* выведет:
Array
(
    [38] => Array
        (
            [a:0:{}] => Array
                (
                    [price_id] => 39
                    [count] => 1
                    [is_file] => 0
                )
        )
    [49] => Array
        (
            [a:0:{}] => Array
                (
                    [price_id] => 60
                    [count] => 1
                    [is_file] => 0
                )
        )
)
```

```

)*/

// запрашиваем количество товара ID=38 в списке пожеланий
echo $this->diafan->_wishlist->get(38, array(), "count");
// выведет: 1

```

integer **get_count ()** – Возвращает количество товаров в списке пожеланий.

Пример:

```

echo 'В списке пожеланий '.$this->diafan->_wishlist->get_count().' товара';
// выведет: В списке пожеланий 2 товара

```

void **set ([mixed \$value = array()], [integer \$id = 0], [mixed \$param = false], [mixed \$additional_cost = false], [string \$name_info = "])** – Записывает данные в список пожеланий.

- mixed \$value: данные
- integer \$id: номер товара
- mixed \$param: характеристики товара, учитываемые в заказе
- mixed \$additional_cost: сопутствующие услуги
- string \$name_info: тип информации (count - количество, is_file - это товар-файл)

Пример:

```

// обновляем количество товара ID=38 в списке пожеланий
// или добавляем его туда
$this->diafan->_wishlist->set(3, 38, array(), "count");
if($err = $this->diafan->_wishlist->set($wishlist, 38, array()))
{
    echo 'Ошибка: '.$err;
}

// обновляем количество товара и пометку, что товар является файлом
// или добавляем его туда
$wishlist = array(
    "count" => 3,
    "is_file" => true,
);
if($err = $this->diafan->_wishlist->set($wishlist, 38, array()))
{
    echo 'Ошибка: '.$err;
}

// удаляем товар ID=38 из списка пожеланий
$this->diafan->_wishlist->set(0, 38, array(), "count");

// очищаем список пожеланий
$this->diafan->_wishlist->set();

```

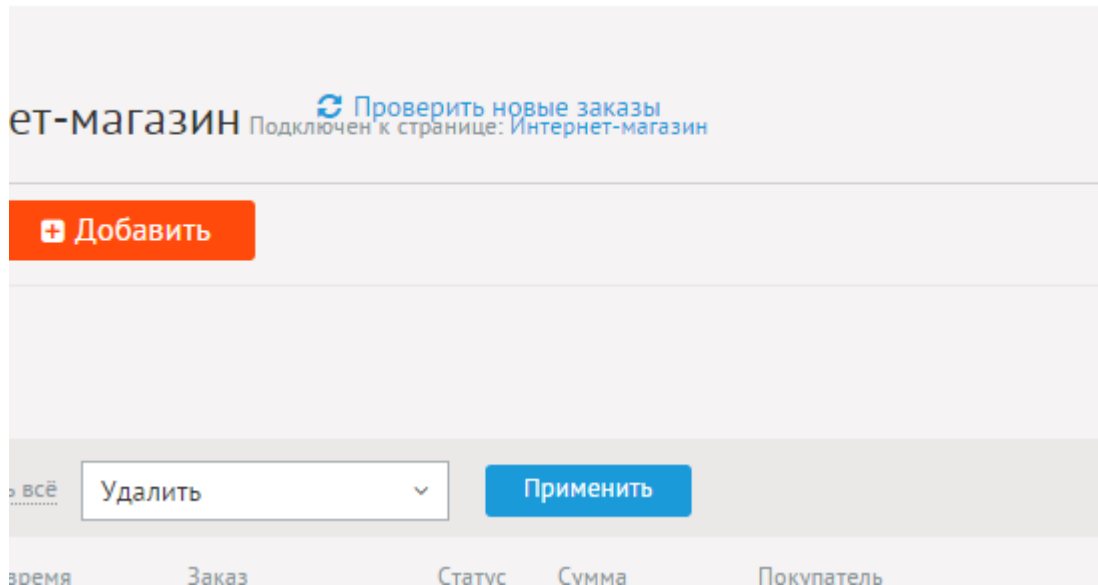
void **write ()** – Записывает информацию в хранилище.

Пример:

```
// очищаем список пожеланий
$this->diafan->_wishlist->set();

// записываем данные, установленные функцией set()
$this->diafan->_wishlist->write();
```

Заказы



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

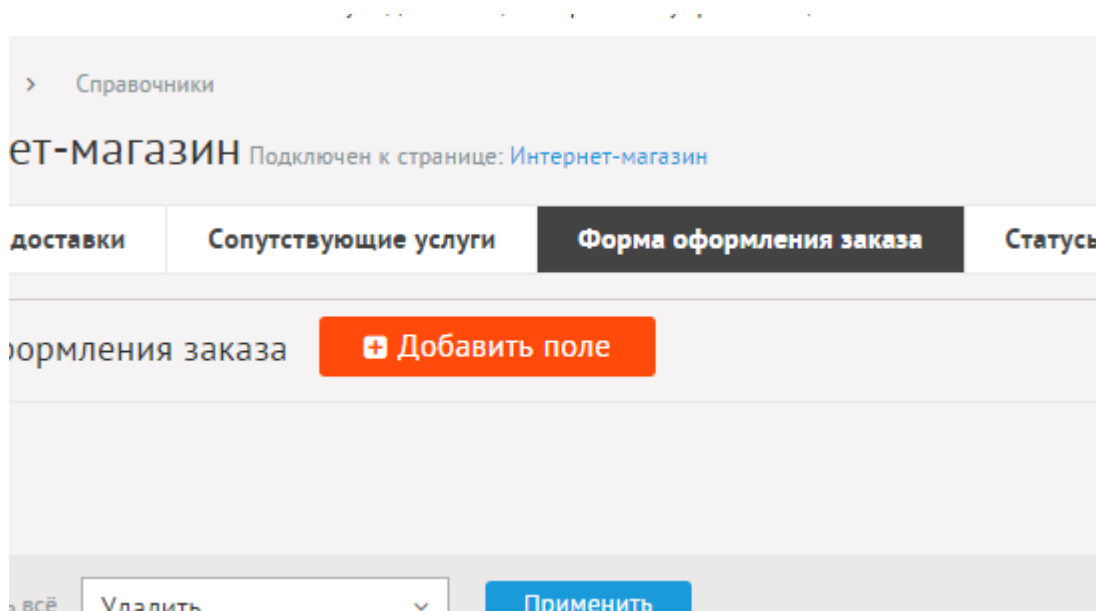
- **Дата** – дата создания заказа;
- **Номер заказа** (в виде ссылки на подробную информацию);
- **Статус** – статусы заказа редактируются в [отдельном интерфейсе](#);
- **Пользователь**, добавивший заказ (в виде ссылки на подробную информацию о пользователе или надпись «без регистрации», если пользователь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

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

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.



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,
 - Телефон,
 - 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.
- **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.

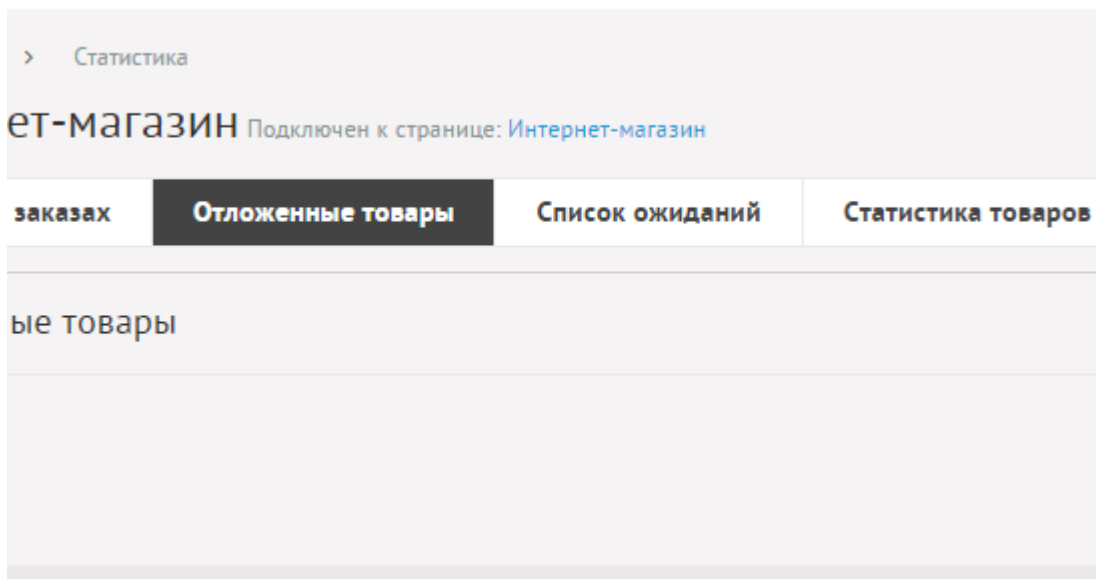
Отчеты

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

ет-магазин Подключен к странице: Интернет-магазин	
заказах	Отложенные товары
Список ожиданий	
Статистика товаров	
Товар	
09:53	Рюкзак «Контур 50» 71041 Цвет: Лесная чаща
09:53	Спальный мешок «Антрим» 34013 Цвет: Зеленый

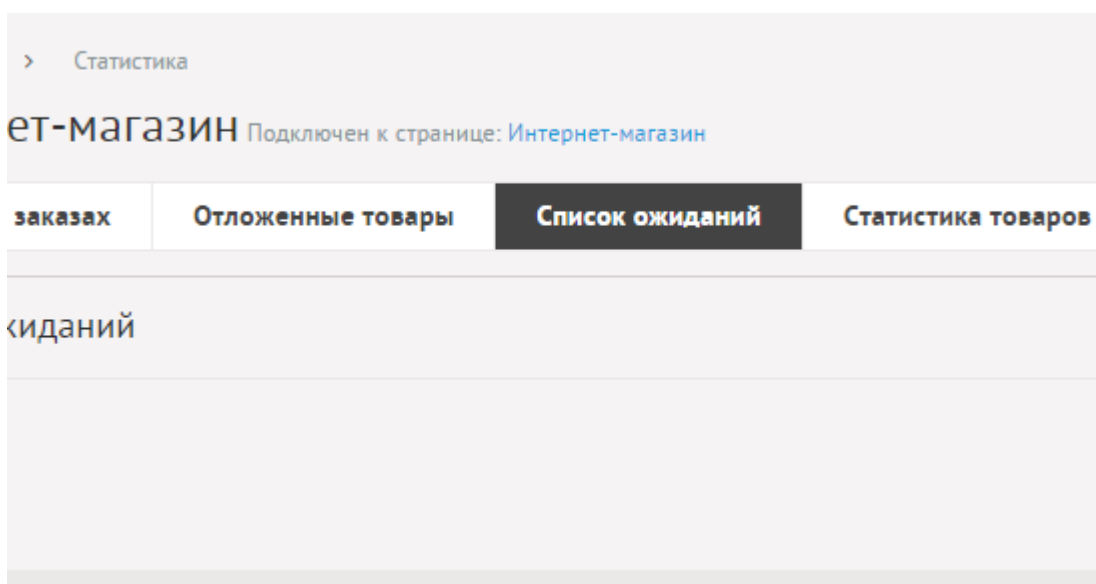
Список пожеланий

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

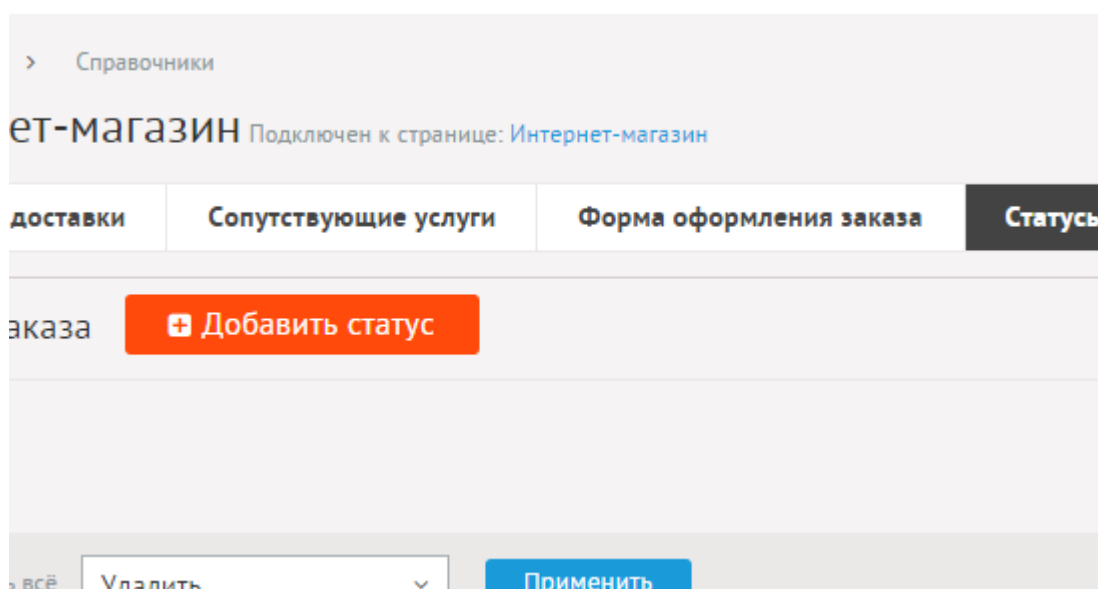


Список ожиданий

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



Статусы 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,
 - отмена заказа,
 - выполнение.
- **Цвет** – Цвет, которым будет выделен статус в списке заказов и в личном кабинете пользователя. Пример: red или #ff0000.
- **Списание товара** – При установке статуса происходит уменьшение количества товара, указанного в заказе, если товары еще не списаны. Если значение не установлено, то при установке статуса списание товаров будет отменено, количество товара на складе увеличиться.
- **Отправлять уведомление пользователю о смене статуса.**
- **Сортировка: установить перед** – Редактирование порядка следования поля в списке.

Скидки

> Каталог > Скидки

Интернет-магазин Подключен к странице: Интернет-магазин

Скидки Производители Категории Характеристики Импорт

+ Добавить скидку

Сделать активным Применить

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

> Каталог > Скидки

Интернет-магазин Подключен к странице: Интернет-магазин

Скидки Производители Категории Характеристики Импорт

+ Добавить скидку

Сделать активным Применить

Скидка 10% действует на все категории и товары

Скидки имеют следующие характеристики:

Размер скидки

- **Активировать скидку.**
- **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.
- **Комментарий для администратора** – Поле видно только администратору.
- **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.
- **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.

The screenshot shows a web interface for an 'Интернет-магазин' (Internet shop). At the top, there is a breadcrumb trail: '> Справочники'. Below it, the text 'ет-магазин' is followed by 'Подключен к странице: Интернет-магазин'. A navigation bar contains four tabs: 'доставки', 'Сопутствующие услуги', 'Форма оформления заказа', and 'Статусы'. Below the tabs is a large orange button with a plus icon and the text 'Добавить валюту'. At the bottom, there is a dropdown menu with 'Удалить' selected and a blue button labeled 'Применить'.

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.

Способы доставки

Неограниченное добавление способов доставки.

Характеристики способов доставки:

Сопутствующие услуги

Сопутствующие услуги можно выбрать при оформлении заказа.

> Справочники

ет-магазин Подключен к странице: Интернет-магазин

доставки	Сопутствующие услуги	Форма оформления заказа	Статусы
<p>сопутствующие услуги + Добавить</p>			
<p> <input type="text" value="поиск по всем"/> <input type="button" value="Сделать активным"/> <input type="button" value="Применить"/> </p>			

Характеристики сопутствующих услуг:

> Справочники > Сопутствующие услуги

ет-магазин Подключен к странице: Интернет-магазин

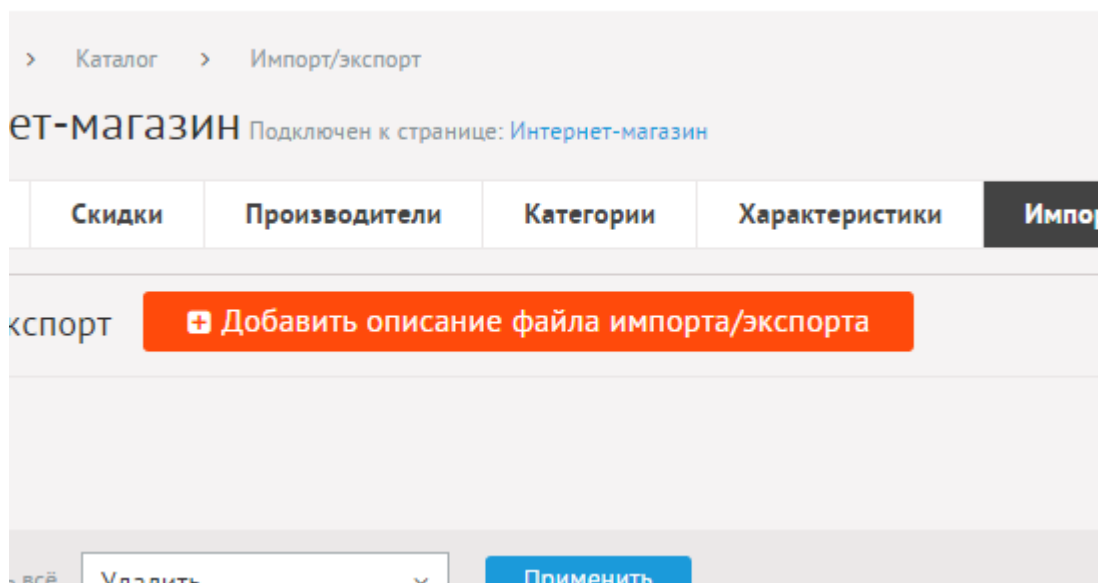
доставки	Сопутствующие услуги	Форма оформления заказа	Статусы
<p>Добавить</p>			
<p>Назначение услуг ?</p> <p> <input type="text" value="внутренняя упаковка"/> </p>			

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.
- **Прикрепить услугу к.**
- **Категории** – Услуга будет выводиться для возможности подключения в товарах выбранных категорий. Чтобы выбрать несколько категорий, удерживайте CTRL.
- **Фиксированная стоимость.**
- **Процент от суммы** – Стоимость услуги составляет процент от стоимости, при этом фиксированная стоимость не учитывается.
- **Бесплатно от суммы** – Стоимость, при которой данная услуга осуществляется бесплатно.

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

Импорт/экспорт

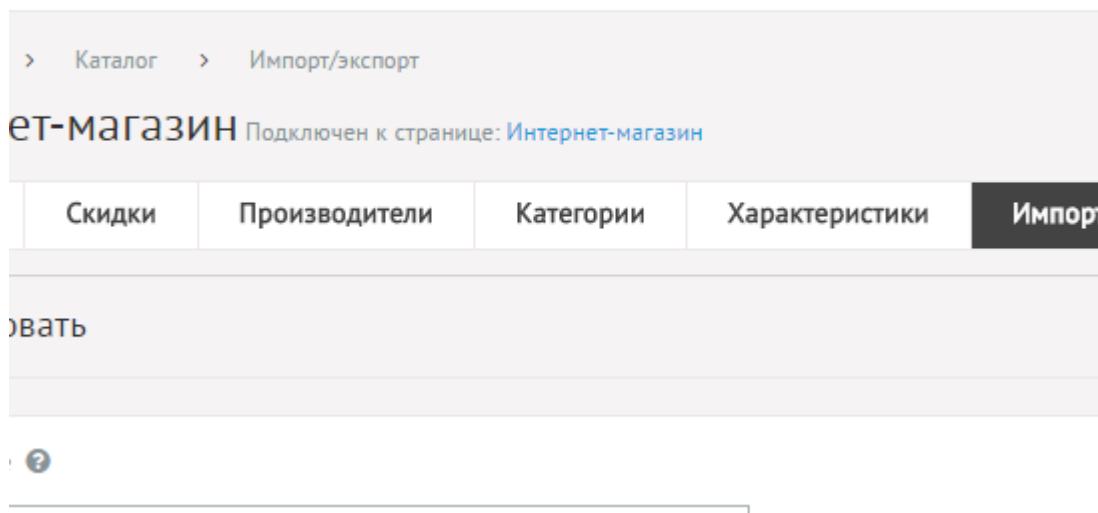


Для импорта и экспорта товаров используется формат [CSV](#).

Прежде чем начать импорт или экспорт Вашего каталога товаров, в DIAFAN.CMS нужно описать содержимое Ваших CSV файлов.

Нажмите на кнопку «Добавить описание файла импорта/экспорта», или используйте один из предварительно созданных описаний (при инсталляции системы создаются два описания, «Импорт товаров» и «Импорт категорий»), Вы можете отредактировать их как угодно.

[Открыть руководство пользователя](#)



Файл имеет следующие характеристики:

Основные:

- **Название** – Краткое описание файла импорта (например, «Импорт товаров», «Импорт цен» и т. д.).
- **Тип** – Возможные значения:
 - Товары,
 - Категории,
 - Производители.
- **Удалять неописанные в файле импорта записи** – Если Вы загружаете список новых товаров или категорий, и то, что уже есть на сайте не нужно, следует отметить эту опцию. На сайт импортируются новые товары из файла CSV, а все уже существующие товары или категории будут удалены, за исключением 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 на большинстве хостингов ограничено, из-за чего скрипт может не успеть обработать весь файл за одну итерацию, если он объемный. Поэтому файл обрабатывается частями, а величину итерации можно задать этим параметром.
- **Разделитель данных в строке** – Разделитель ячеек в строке файла CSV. По умолчанию ;
- **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. Редко используется, это не обязательный параметр.
- **Кодировка** – Кодировка данных в файле CSV. Часто cp1251 или utf8. По умолчанию из Excell файлы CSV выходят в кодировке cp1251
- **Разделитель данных внутри поля** – В некоторых полях (ячейках) может быть несколько данных (например, значение характеристики с типом «список с выбором нескольких значений» или несколько имен изображений для одного товара). В этом случае данные должны быть разделены этим разделителем.
- **Первая строка – названия полей** – Если отмечено, то добавляется описание полей в первой строке файла при экспорте, а при импорте первая строка игнорируется.
- **Сортировка: установить перед** – Редактирование порядка следования категории в списке. Поле доступно для редактирования только для категорий, отображаемых на сайте.

В описании импорта есть конструктор полей, с помощью которого нужно создать поля, отражающие колонки в таблице CSV, содержащиеся в файле.

[Каталог](#) > [Импорт/экспорт](#) > [Импорт категорий](#)

ет-магазин Подключен к странице: [Интернет-магазин](#)

Скидки	Производители	Категории	Характеристики	Импорт
Категорий изменить + Добавить поле				
<div> Всё <input type="text" value="Вплотить"/> ▼ Применить </div>				

Поля файла имеют следующие характеристики:

[Каталог](#) > [Импорт/экспорт](#) > [Импорт товаров](#)

ет-магазин Подключен к странице: [Интернет-магазин](#)

Скидки	Производители	Категории	Характеристики	Импорт
<div> Всё <input type="text" value="Вплотить"/> ▼ Применить </div>				
<div> Всё <input type="text" value="Вплотить"/> ▼ Применить </div>				

?

Основные:

- **Название** – Название поля для импорта, необходимо только для наглядности в списке полей конструктора форм в административной части сайта.
- **Тип** – Значение или свойство товара, куда будет импортироваться данное поле.

Возможные значения:

- Идентификатор (уникальный код),
- Название – поле «Название»,
- Артикул – используется у товаров,
- Единица измерения – 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,
- Производитель – идентификатор производителя из файла импорта производителей, используется у товаров. Чтобы производитель для товара определилась правильно, нужно сначала импортировать производителей,
- 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». Только имена, вида img123.jpg,
- 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,
- Редирект – ссылка относительно корня сайта, без слеша в начале; если указан «Дополнительный разделитель», то можно указать код редиректа,
- Канонический тег – полная ссылка,
- 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,
- Выгружать в Google Merchant – значения 1|0|true|false,
- Значения полей для Google Merchant – 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,
- Changefreq – значения,
- Priority – значения 0 - 1,
- Номер для сортировки – товары сортируются по убыванию, категории и производители по возрастанию,
- Редактор – id пользователя на сайте,
- Шаблон сайта – файл из папки themes,
- Шаблон модуля (*modules/shop/views/shop.view.шаблон.php*),
- Шаблон модуля для списка товаров (*modules/shop/views/shop.view.шаблон.php*),

- Шаблон страницы элемента (*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,
- Вес,
- Длина,
- Ширина,
- Высота.
- **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», «Производитель» и «Идентификатор связанных товаров».
 - собственное значение – при первом импорте все товары/категории/производители добавляются в базу, идентификатор запишется в поле `import_id`. При последующем импорте товары/категории/производители 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 и производителей, только для типов «Категория», «Производитель» 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.
- **Адрес файлов для загрузки** – Может быть вида `pictures` (тогда будет использоваться локальная папка текущего сайта `http://site.ru/pictures/`). Или в виде полного онлайн пути `http://anysite.ru/pictures/` (для .рф доменов в PUNY-формате). К этому пути при импорте добавятся имена изображений из импортируемого файла CSV. Используется только для типов полей «Имена изображений» и «Дополнительная характеристика» с типами «Изображения» и «Файлы».
- **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 старую цену** – 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».
- **Указывать валюту** – Значение следует сразу за старой ценой или ценой или количеством (если не отмечены опции «Указывать количество» и «Указывать старую цену») 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.

- **Дополнительный разделитель** – Дополнительный разделитель полей в ячейке. В строке данные по умолчанию делятся разделителем из глобальных настроек импорта (по умолчанию |). Если это поле заполнить, то для типа «Редирект» можно указать через разделитель код редиректа, а для типа «Имена изображений» значения alt и title для изображений.
- **Категория** – Файл импорта.
- **Сортировка: установить после** – Изменить положение текущего поля среди других полей. В списке можно сортировать поля простым перетаскиванием мыши.

Под конструктором полей импорта есть форма для загрузки файла импорта CSV и ссылка «Экспорт» для скачивания файла экспорта. При импорте все данные проверяются на валидность, и в случае неправильного формата или некорректных значений выводится лог ошибок. При экспорте скачивается файл в товарами, которые есть на текущий момент в каталоге товаров, в формате CSV с теми полями, что описаны в конструкторе.

[Примеры импорта товаров и категорий](#)

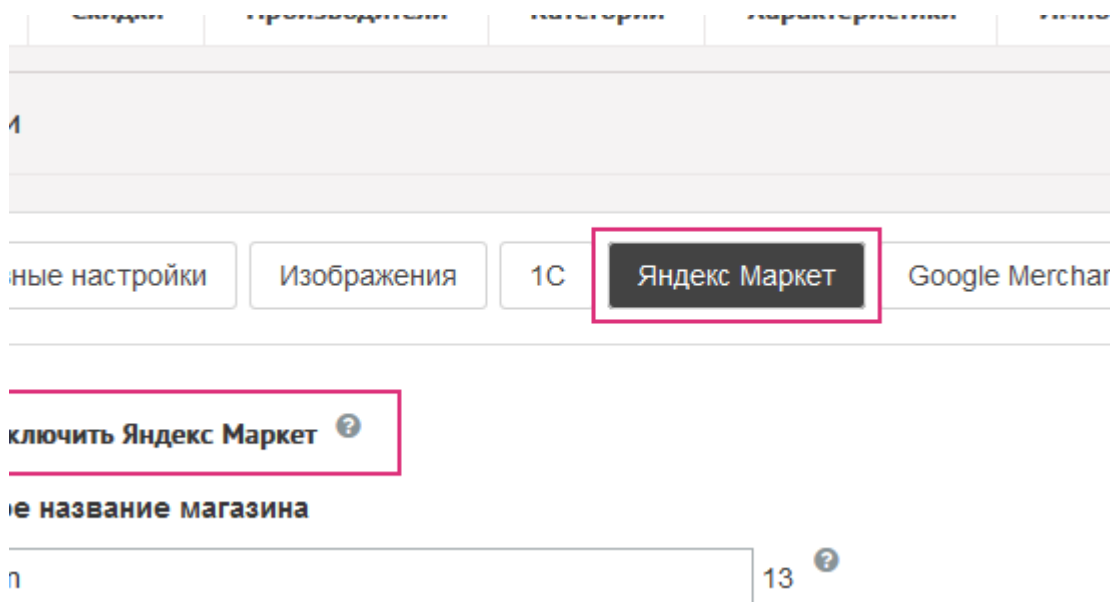
Экспорт товаров для Яндекс.Маркет

Экспорт товаров в формате YML для Яндекс.Маркет

Выгрузка товаров в формате YML всегда доступна на сайте по адресу

<http://site.ru/modules/shop/shop.yandex.php>

*при включённой опции «Подключить Яндекс Маркет» в настройках магазина



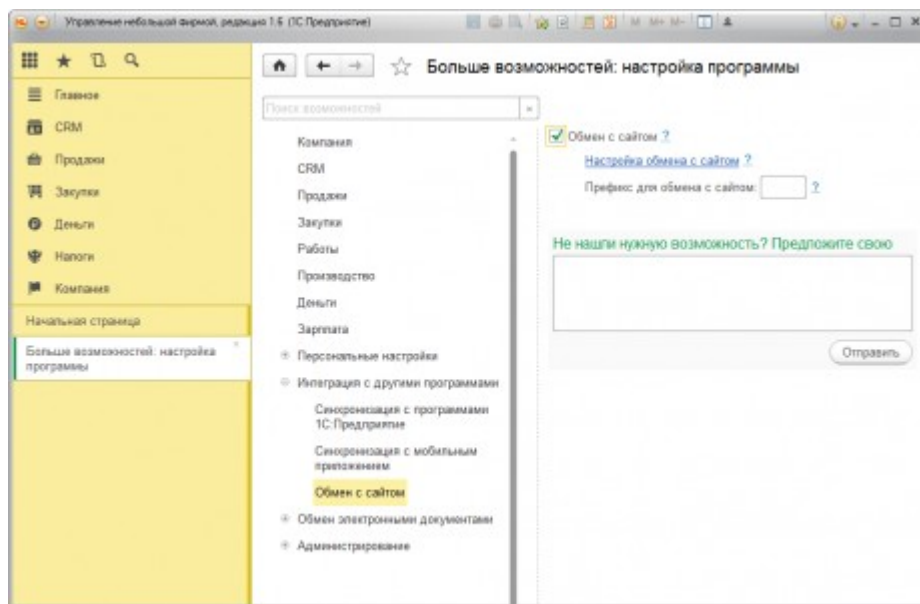
Интеграция с системой 1С:Предприятие

DIAFAN.CMS интегрирована с системой 1С:Предприятие и имеет [официальный сертификат 1С:Совместимо](#). Важно понимать, что обмен данными между любой CMS на любом сайте и программой 1С работает по единому стандарту на основе открытого протокола обмена данными, разработанного компанией 1С. Если у Вас в программе 1С есть модуль обмена данными с сайтом, значит Ваш сайт поддерживает синхронизацию, независимо от прикладного решения самой программы (1С:Управление фирмой, 1С:Комплексная автоматизация или 1С:Управление торговлей).

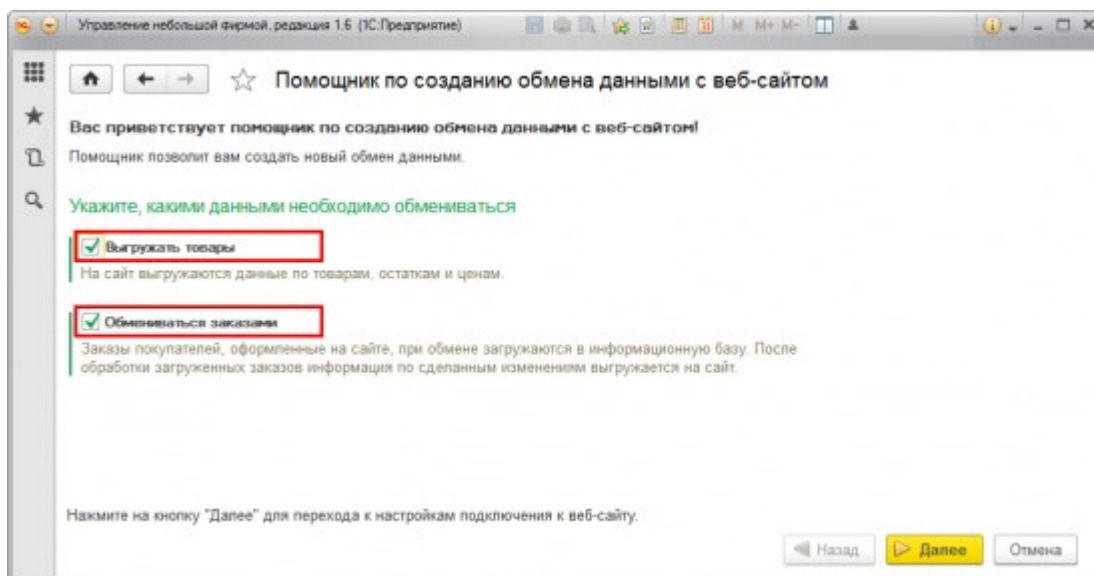
Согласно протоколу 1С, программа является ведущей, а сайт ведомым. То есть, при синхронизации программы 1С и сайта, и выгрузке и обновлении номенклатуры, любые данные из 1С принимаются приоритетными и затирают существующее на сайте. То есть, при синхронизации с 1С правильно вести весь учет именно в программе, обновляя всё данные, а на сайт все изменения будут попадать автоматически.

Для синхронизации модуля «Магазин» DIAFAN.CMS и системы 1С:Предприятие выберите

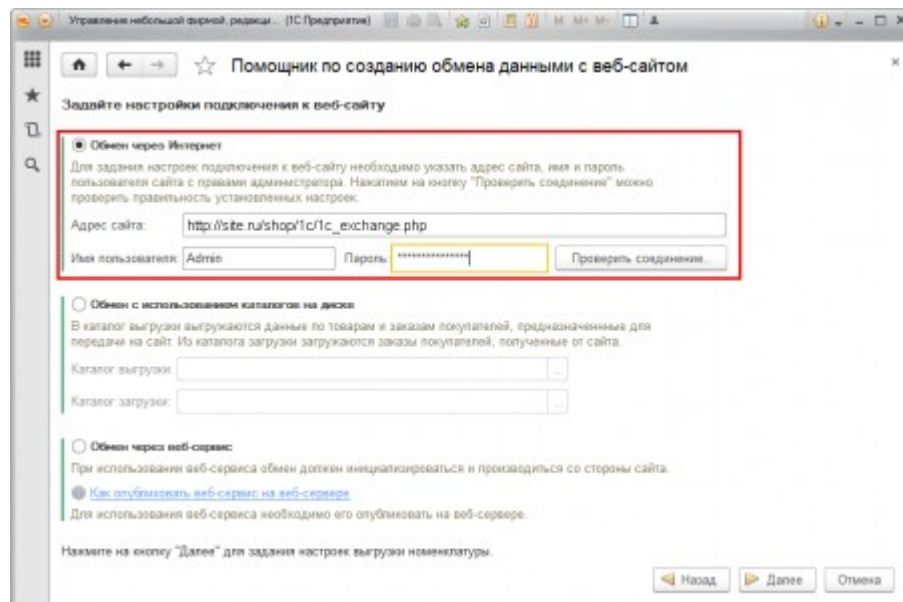
пункт *Компания - Интеграция с др. программами - Обмен с сайтом* в программе 1С и установите опцию *Обмен с сайтом*. После этого станет доступна ссылка *Настройка обмена с сайтом*.



На первом шаге настройки отметьте *Выгружать товары* и/или *Обмениваться заказами*.



На втором шаге отметьте *Обмен через интернет* и укажите адрес для подключения к сайту, имя и пароль пользователя.



Заполните поля:

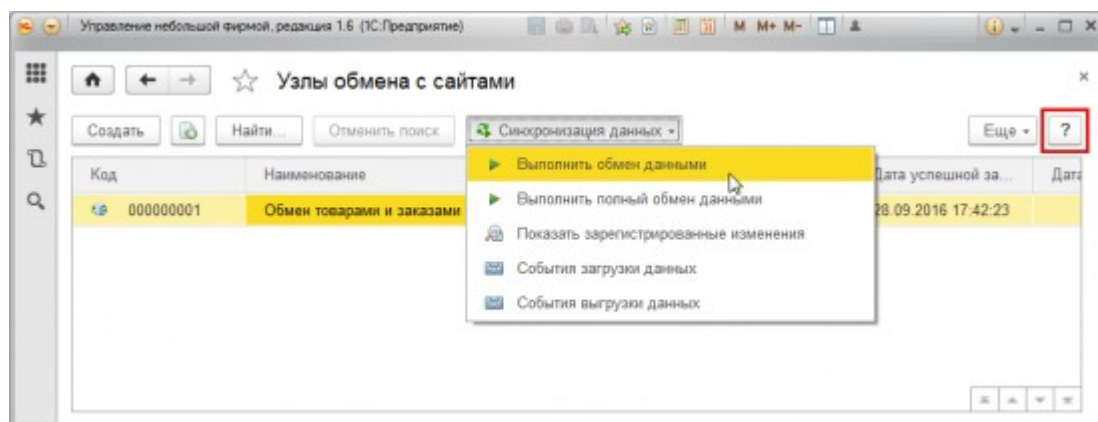
адрес сайта: http://site.ru/shop/1c/1c_exchange.php подставив вместо site.ru адрес своего сайта.

Внимание! Не пытайтесь открыть этот адрес в браузере! Скрипт ответит только программе 1С по её запросу с авторизационными данными, которые будут внесены в настройки.

Пользователь: логин пользователя из панели управления сайтом, у которого есть доступ к модулю «Магазин», часто **admin** **Пароль:** пароль этого пользователя

На следующих шагах необходимо задать настройки для выгрузки номенклатуры и их изображений, настройки для обмена заказами, а также настройку расписания для автоматического обмена.

После создания Узла обмена, Вы можете запускать синхронизацию вручную или она будет запускаться автоматически по расписанию, согласно настройкам.



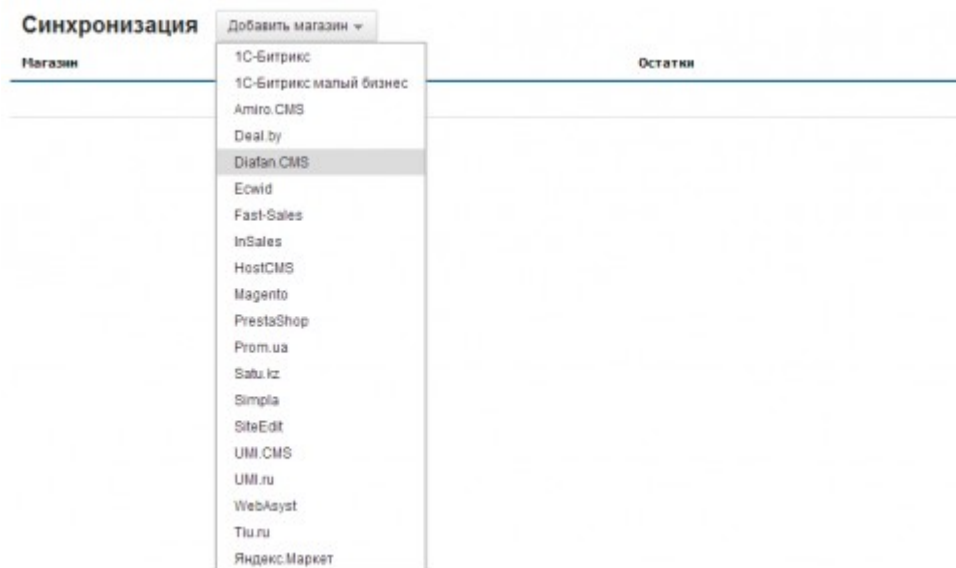
Интеграция разрабатывалась в соответствии с [протоколом обмена между системой «1С:Предприятие» и сайтом](#). Обмен происходит по следующим направлениям:

- 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 МойСклад: **`http://site.ru/shop/1c/1c_exchange.php`** подставив вместо site.ru адрес своего сайта. 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

Настройки 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 о кодировке](#)

Заказы

☒ Загружать каждые минут

☐ Копировать номера заказов из магазина

Добавлять к номеру префикс:

☐ Резервировать товары

Склад:

* Организация: ж

☐ Отправлять статусы заказов в 1С

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.

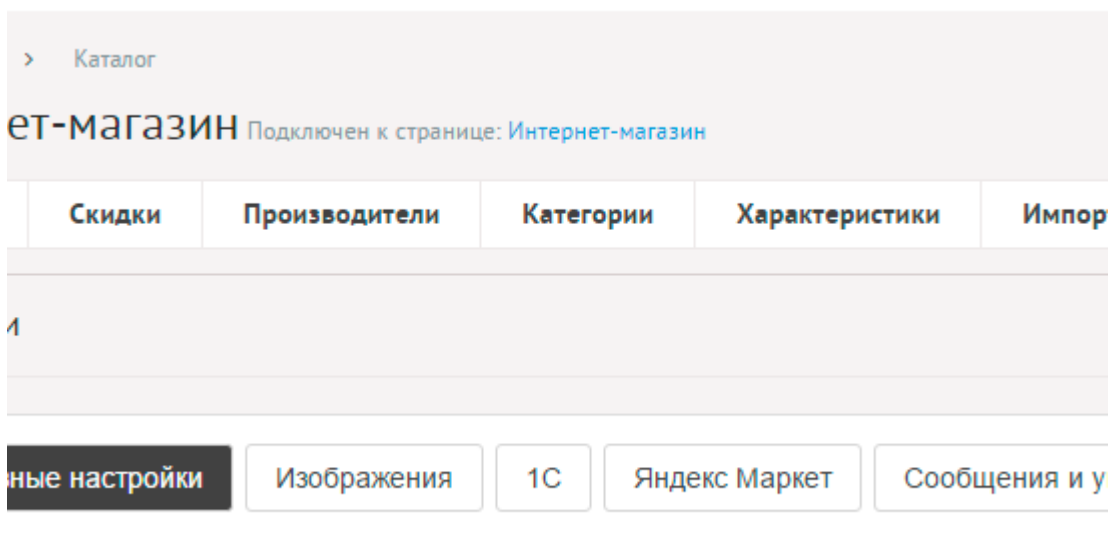
Из группы товаров — группа товаров, для товаров из которой будут выгружаться остатки. Если группа не указана, то будут выгружены остатки для всех товаров.

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.



Основные настройки

- **Основные.**
- **Запретить покупать товары** – Чтобы запретить пользователям покупать товары и использовать сайт как интернет-магазин 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 выполнении заказа (статус заказа с товаром «Выполнен») количество будет минусоваться.
- **Скрывать отсутствующие товары** – Если отмечена, товары с количеством 0 или товары, помеченные опцией «Товар временно отсутствует» исчезнут из списков товаров. Страницы таких товаров по-прежнему будут доступны.
- **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.
- **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

- **Включить «Заказать в один клик»** – На странице товара появится форма быстрой покупки товара. Поля для формы выводятся опцией «Использовать в форме быстрого заказа».
- **Название основной валюты** – Название основной валюты сайта. Для добавления дополнительных валют, воспользуйтесь интерфейсом «Справочники» — «Валюты». Параметр имеет разные значения для разных языковых версий сайта.
- **Формат цены** – Возможность указать формат отображения цены (количество знаков после запятой, разделитель целых, разделитель десятков).
- **Налог** – Название налога на добавленную стоимость.
- **Величина налога, %** – Если 0, то налог нигде не отображается.
- **Минимальная сумма заказа** – Оформление заказа возможно при достижении минимальной суммы заказа. Если минимальная сумма заказа меньше или равна нулю, данное ограничение не действует.
- **Максимальная сумма заказа** – Оформление заказа возможно при непревышении максимальной суммы заказа. Если максимальная сумма заказа меньше или равна нулю, данное ограничение не действует.
- **Определение суммы заказа с учетом скидок** – При определении минимальной или максимальной суммы заказа учитываются возможные скидки.
- **Определение суммы заказа с учетом сопутствующих услуг** – При определении минимальной или максимальной суммы заказа учитываются возможные сопутствующие услуги.
- **По окончании оформления заказа** – Страница сайта, на которую попадает покупатель после успешного оформления заказа.
- **Количество товаров на странице** – Количество одновременно выводимых товаров в списке.
- **Включить «Показать ещё»** – На странице товаров появится кнопка «Показать ещё». Увеличивает количество одновременно выводимых товаров в списке.
- **Выводить на первой странице весь список товаров** – По умолчанию на первой странице выводится список категорий с несколькими товарами. Параметр появляется, если отмечена опция «Использовать категории».
- **Сортировка товаров** – Выбранная сортировка будет применена и в административной панели и на сайте. Возможные значения:
 - ручная сортировка,
 - по дате добавления: от нового к старому,
 - по дате добавления: от старого к новому,
 - по алфавиту.
- **В блоке похожих товаров связь двусторонняя** – Если отметить, то при назначении товару А похожего товара Б, у товара Б автоматически станет похожим товар А.
- **Категории.**
- **Использовать категории** – Позволяет включить/отключить категории товаров.
- **Количество категорий на странице** – Количество одновременно выводимых категорий в списке на первой страницы модуля. Параметр появляется, если

отмечена опция «Использовать категории».

- **Количество товаров в списке категорий** – Для первой страницы магазина, где выходят по несколько товаров из всех категорий. 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».
- **Количество товаров в списке вложенной категории** – Для первой страницы модуля и для страницы категории. Параметр появляется, если отмечена опция «Использовать категории».
- **Поиск.**
- **Искать по цене** – Параметр позволяет выводить в блоке поиска по товарам поиск по цене.
- **Искать по артикулу** – Параметр позволяет выводить в блоке поиска по товарам поиск по артикулу.
- **Искать по производителям** – Параметр позволяет выводить в блоке поиска по товарам список производителей для выбора.
- **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»).
- **Скидки.**
- **Код купона не уникален** – Параметр позволяет вводить одинаковый код для разных купонов. При вводе кода активируются все купоны с таким кодом.
- **Сервис.**
- **Оптимизировать таблицу БД цены товаров** – Если у товара записано две цены с одинаковыми характеристиками, то одна из них (более ранняя запись) будет удалена.
- **Подключения.**
- **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»](#).
- **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».
- **Автогенерация для SEO.**
- **Шаблон для автоматического генерирования Title** – Если шаблон задан и для товара не прописан заголовок *Title*, то заголовок автоматически генерируется по шаблону. В шаблон можно добавить:
 - %name – название,
 - %category – название категории,
 - %parent_category – название категории верхнего уровня,
 - %article – артикул (SEO-специалисту).

Параметр имеет разные значения для разных языковых версий сайта.

- **Шаблон для автоматического генерирования Keywords** – Если шаблон задан и для товара не заполнено поле *Keywords*, то поле *Keywords* автоматически генерируется по шаблону. В шаблон можно добавить:
 - %name – название,
 - %category – название категории,
 - %parent_category – название категории верхнего уровня,
 - %article – артикул (SEO-специалисту).

Параметр имеет разные значения для разных языковых версий сайта.

- **Шаблон для автоматического генерирования Description** – Если шаблон задан и для товара не заполнено поле *Description*, то поле *Description* автоматически генерируется по шаблону. В шаблон можно добавить:
 - %name – название,
 - %category – название категории,
 - %parent_category – название категории верхнего уровня,
 - %anons – краткое описание,
 - %article – артикул (SEO-специалисту).

Параметр имеет разные значения для разных языковых версий сайта.

- **Шаблон для автоматического генерирования Title для категории** – Если шаблон задан и для категории не прописан заголовок *Title*, то заголовок автоматически генерируется по шаблону. В шаблон можно добавить:
 - %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 и для категории не заполнено поле *Keywords*, то поле *Keywords* автоматически генерируется по шаблону. В шаблон можно добавить:

- %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 и для категории не заполнено поле *Description*, то поле *Description* автоматически генерируется по шаблону. В шаблон можно добавить:

- %name – название категории,
- %parent – название категории верхнего уровня,
- %anons – краткое описание (SEO-специалисту).

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

- **Шаблон для автоматического генерирования Title для производителей** – Если шаблон задан и для производителя не прописан заголовок *Title*, то заголовок автоматически генерируется по шаблону. В шаблон можно добавить:

- %name – название производителя,
- %page – страница (текст можно поменять в интерфейсе «Языки сайта» – «Перевод интерфейса»)

(SEO-специалисту). Параметр имеет разные значения для разных языковых версий сайта.

- **Шаблон для автоматического генерирования Keywords для производителей** – Если шаблон задан и для производителя не заполнено поле *Keywords*, то поле *Keywords* автоматически генерируется по шаблону. В шаблон можно добавить:

- %name – название производителя

(SEO-специалисту). Параметр имеет разные значения для разных языковых версий сайта.

- **Шаблон для автоматического генерирования Description для производителей** – Если шаблон задан и для производителя не заполнено поле *Description*, то поле *Description* автоматически генерируется по шаблону. В шаблон можно добавить:

- %name – название производителя

(SEO-специалисту). Параметр имеет разные значения для разных языковых версий сайта.

- **Оформление.**

- **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 элементов в списке** – По умолчанию

- modules/shop/views/shop.view.rows.php*. Параметр для разработчиков! Не устанавливайте, если не уверены в результате. Значение параметра важно для AJAX.
- **Шаблон для первой страницы модуля (если подключены категории)** – По умолчанию *modules/shop/views/shop.view.fist_page.php*. Параметр для разработчиков! Не устанавливайте, если не уверены в результате.
 - **Шаблон для элементов в списке первой страницы модуля (если подключены категории)** – По умолчанию *modules/shop/views/shop.view.fist_page.php*. Параметр для разработчиков! Не устанавливайте, если не уверены в результате. Значение параметра важно для AJAX.
 - **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 товаров производителя** – По умолчанию, *modules/shop/views/shop.view.list.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 элементов списка с одинаковой характеристикой** – По умолчанию, *modules/shop/views/shop.view.rows.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.
 - **Шаблон элементов в списке для поиска элементов** – По умолчанию *modules/shop/views/shop.view.rows.php*. Параметр для разработчиков! Не устанавливайте, если не уверены в результате. Значение параметра важно для AJAX.
 - **Индексирование для карты сайта** – При изменении настроек, влияющих на отображение страницы, модуль автоматически переиндексируется для карты сайта *sitemap.xml*.

Изображения

- **Использовать изображения для товаров** – 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 изображения для производителей** – Позволяет включить/отключить загрузку изображений для производителей.
- **Генерировать размеры изображений для производителей** – Размеры изображений, заданные в модуле «Изображения» и тег латинскими буквами для подключения изображения на сайте. Параметр выводится, если отмечена опция «Использовать изображения для производителей».
- **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».

1С

- **Активировать новые товары, категории и производителей после синхронизации** – Если отметить, то товары, категории и производители, добавленные из системы 1С:Предприятие, будут сразу показаны на сайте.
- **Выгружать все заказы, включая ранее выгруженные** – Если не отмечено, то с сайта выгружаются только заказы старше даты последней выгрузки.
- **Вести лог ошибок** – Если не отмечено, то ниже будет выведен лог ошибок выгрузки.

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,
 - BYN,
 - 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».
- **Подключить Яндекс Быстрый заказ** – Позволяет заполнять форму оформления заказа данными, предзаполненными в [системе «Яндекс Быстрый заказ»](#).

Google Merchant

- **Подключить Google Merchant** – Если отметить, по адресу <http://www.site.ru/shop/google/> будет активен файл с импортом товаров для системы Google Merchant в формате XML. Все подробности и требования к магазинам смотрите на [сайте Google Merchant](#).
- **Валюта** – Валюта для системы Google Merchant. Возможные значения:
 - RUB,

- USD,
- EUR,
- ARS,
- AUD,
- BRL,
- CAD,
- CLP,
- COP,
- CZK,
- DKK,
- HKD,
- INR,
- IDR,
- JPY,
- MYR,
- MXN,
- NZD,
- NOK,
- PHP,
- PLN,
- SGD,
- ZAR,
- SEK,
- CHF,
- TWD,
- TRY,
- AED,
- GBP.

Параметр появляется, если отмечена опция «Подключить Google Merchant».

- **Единица измерения размеров (ширина, длина, высота)** – Возможные значения:
 - сантиметр,
 - дюйм.

Параметр появляется, если отмечена опция «Подключить Google Merchant».

- **Единица измерения веса** – Возможные значения:
 - килограмм,
 - фунт,
 - унция,
 - грамм.

Параметр появляется, если отмечена опция «Подключить Google Merchant».

- **Выгружать категории в Google Merchant** – Позволяет выбрать товары каких категорий выгружать в Google Merchant: все или только помеченные (появляется галочка при редактировании категории). Возможные значения:
 - все,
 - только помеченные.

Параметр появляется, если отмечена опция «Подключить Google Merchant».

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

Параметр появляется, если отмечена опция «Подключить Google Merchant».

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:

Товар поступил на склад

- **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. Можно добавлять:
 - %id – номер заказа.

Параметр имеет разные значения для разных языковых версий сайта.

- **Сообщение о принятии платежа** – 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.

- **Сообщение об ошибке платежа** – 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),
- %id – номер заказа.

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»,
- %fio – имя пользователя,
- %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),
- %id – номер заказа,
- %message – поля формы «Оформление заказа».

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»,
- %fio – имя пользователя, совершившего заказ,
- %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».
- **Тема письма пользователю о купленных файлах** – Можно добавлять:
 - %title – название сайта,

- %url – адрес сайта (например, site.ru),
- %id – номер заказа.

Пример:

Вы оформили заказ %id на сайте %title (%url)

- **Текст письма 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!

- **Тема письма пользователю о брошенной корзине** – Можно добавлять:
 - %title – название сайта,
 - %url – адрес сайта (например, site.ru).
- **Текст письма пользователю о брошенной корзине** – Можно добавлять:
 - %title – название сайта,
 - %url – адрес сайта (например, site.ru),
 - %goods – ссылки на товары,
 - %link – ссылка на оформление заказа.

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:

- **defer** – маркер отложенной загрузки шаблонного тега: **event** – загрузка контента только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** – загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка контента только по желанию пользователя;
- **defer_title** – текстовая строка, выводимая на месте появления загружаемого контента с помощью отложенной загрузки шаблонного тега;
- **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 шаблоне тега можно получить значение любого атрибута через переменную `$result["attributes"]["название атрибута"]`.

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

Атрибуты:

- **defer** – маркер отложенной загрузки шаблонного тега: **event** – загрузка контента только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** – загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка контента только по желанию пользователя;
- **defer_title** – текстовая строка, выводимая на месте появления загружаемого контента с помощью отложенной загрузки шаблонного тега;
- **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

В шаблоне тега можно получить значение любого атрибута через переменную `$result["attributes"]["название атрибута"]`.

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. Можно указать отрицательное значение, тогда будут исключены товары из указанного раздела. По умолчанию выбираются все страницы;
- **cat_id** – категории товаров, если в настройках модуля отмечено «Использовать категории». Можно указать отрицательное значение, тогда будут исключены товары из указанной категории. Идентификаторы категорий перечисляются через запятую.

Можно указать значение **current**, тогда будут показаны товары из текущей (открытой) категории магазина или из всех категорий, если ни одна категория не открыта. По умолчанию категория не учитывается, выводятся все товары;

— **ids** – Товары. Можно указать отрицательное значение, тогда будут исключены товары. Идентификаторы товаров перечисляются через запятую. По умолчанию товары не учитываются, выводятся все товары;

— **brand_id** – производители товаров. Можно указать отрицательное значение, тогда будут исключены товары указанного производителя. Идентификаторы производителя перечисляются через запятую. По умолчанию производитель не учитывается, выводятся все товары;

— **sort** – сортировка товаров: по умолчанию как на странице модуля, **date** – по дате, **rand** – в случайном порядке, **price** – по цене, **sale** – по количеству продаж;

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

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

— **param** – значения дополнительных характеристик;

Пример:

Товары обладают следующими характеристиками:

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

Значит значение атрибута *param="3=5&3=6&10>12&16=0"* расшифровывается как товары красного и синего цвета (5 и 6 номер), высотой более 12, не имеющие аналогов. Символы **<** и **>** нужно заменять HTML-мнемониками **<** и **>**.

```
<insert name="show_block" module="shop" param="3=5&3=6&10&gt;12&16=0">
```

Номер (или идентификатор) характеристики можно посмотреть, если подвести курсор к названию характеристики в списке характеристик в административной части. Появится всплывающая подсказка «Редактировать (номер характеристики)».

Номер (или идентификатор) значения характеристики можно посмотреть, если при редактировании характеристики подвести курсора на нужное значение. Появится всплывающая подсказка «ID: номер».

— **hits_only** – выводить только товары с пометкой «Хит»: **true** – выводить только товары с пометкой «Хит», по умолчанию пометка «Хит» будет игнорироваться;

— **action_only** – выводить только товары с пометкой «Акция»: **true** – выводить только

- товары с пометкой «Акция», по умолчанию пометка «Акция» будет игнорироваться;
- **new_only** – выводить только товары с пометкой «Новинка»: **true** – выводить только товары с пометкой «Новинка», по умолчанию пометка «Новинка» будет игнорироваться;
 - **discount_only** – выводить только товары, на которые действует скидка: **true** – выводить только товары, на которые действует скидка, по умолчанию скидка у товаров игнорируется;
 - **only_module** – выводить блок только на странице, к которой прикреплен модуль «Магазин»: **true** – выводить блок только на странице модуля, по умолчанию блок будет выводиться на всех страницах;
 - **tag** – тег, прикрепленный к товарам;
 - **defer** – маркер отложенной загрузки шаблонного тега: **event** – загрузка контента только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** – загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка контента только по желанию пользователя;
 - **defer_title** – текстовая строка, выводимая на месте появления загружаемого контента с помощью отложенной загрузки шаблонного тега;
 - **template** – шаблон тега (файл *modules/shop/views/shop.view.show_block_***template***.php*; по умолчанию шаблон *modules/shop/views/shop.view.show_block.php*).

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

В шаблоне тега можно получить значение любого атрибута через переменную `$result["attributes"]["название атрибута"]`.

show_block module="wishlist" – выводит информацию о товарах в списке пожеланий.

Атрибуты:

- **defer** – маркер отложенной загрузки шаблонного тега: **event** – загрузка контента только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** – загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка контента только по желанию пользователя;
- **defer_title** – текстовая строка, выводимая на месте появления загружаемого контента с помощью отложенной загрузки шаблонного тега;
- **template** – шаблон тега (файл *modules/wishlist/views/wishlist.view.show_block_template.php*; по умолчанию шаблон *modules/wishlist/views/wishlist.view.show_block.php*).

Пример:

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

В шаблоне тега можно получить значение любого атрибута через переменную `$result["attributes"]["название атрибута"]`.

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;
- **defer** – маркер отложенной загрузки шаблонного тега: **event** – загрузка контента только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** – загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка контента только по желанию пользователя;
- **defer_title** – текстовая строка, выводимая на месте появления загружаемого контента с помощью отложенной загрузки шаблонного тега;
- **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

В шаблоне тега можно получить значение любого атрибута через переменную `$result["attributes"]["название атрибута"]`.

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;
- **defer** – маркер отложенной загрузки шаблонного тега: **event** – загрузка контента только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** – загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка контента только по желанию пользователя;
- **defer_title** – текстовая строка, выводимая на месте появления загружаемого контента с помощью отложенной загрузки шаблонного тега;
- **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

В шаблоне тега можно получить значение любого атрибута через переменную `$result["attributes"]["название атрибута"]`.

show_brand – выводит несколько производителей.

Атрибуты:

- **count** – количество выводимых производителей (по умолчанию выводятся все производители);
- **site_id** – страницы, к которым прикреплен модуль. Идентификаторы страниц

перечисляются через запятую. Можно указать отрицательное значение, тогда будут исключены производители из указанного раздела. По умолчанию выбираются все страницы;

— **cat_id** – категории товаров, если в настройках модуля отмечено «Использовать категории». Можно указать отрицательное значение, тогда будут исключены производители из указанной категории. Идентификаторы категорий перечисляются через запятую. Можно указать значение **current**, тогда будут показаны производители из текущей (открытой) категории магазина или из всех категорий, если ни одна категория не открыта. По умолчанию категория не учитывается, выводятся все производители;

— **sort** – сортировка производителей: по умолчанию как на странице модуля, **name** – по имени, **rand** – в случайном порядке;

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

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

— **only_module** – выводить блок только на странице, к которой прикреплен модуль «Магазин»: **true** – выводить блок только на странице модуля, по умолчанию блок будет выводиться на всех страницах;

— **defer** – маркер отложенной загрузки шаблонного тега: **event** – загрузка контента только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** – загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка контента только по желанию пользователя;

— **defer_title** – текстовая строка, выводимая на месте появления загружаемого контента с помощью отложенной загрузки шаблонного тега;

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

Пример:

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

В шаблоне тега можно получить значение любого атрибута через переменную `$result["attributes"]["название атрибута"]`.

show_category – выводит несколько категорий.

Атрибуты:

— **site_id** – страницы, к которым прикреплен модуль. Идентификаторы страниц перечисляются через запятую. Можно указать отрицательное значение, тогда будут

исключены категории из указанного раздела. По умолчанию выбираются все страницы;

- **images** – количество изображений, прикрепленных к категории;
- **images_variation** – тег размера изображений, задается в настройках модуля;
- **only_module** – выводить блок только на странице, к которой прикреплен модуль «Магазин»: **true** – выводить блок только на странице модуля, по умолчанию блок будет выводиться на всех страницах;
- **count_level** – количество уровней;
- **number_elements** – выводить количество товаров в категории: **true** – выводить количество товаров, по умолчанию количество не выводится;
- **defer** – маркер отложенной загрузки шаблонного тега: **event** – загрузка контента только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** – загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка контента только по желанию пользователя;
- **defer_title** – текстовая строка, выводимая на месте появления загружаемого контента с помощью отложенной загрузки шаблонного тега;
- **template** – шаблон тега (файл *modules/shop/views/shop.view.show_category_template.php*; по умолчанию шаблон *modules/shop/views/shop.view.show_category.php*).

Пример:

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

В шаблоне тега можно получить значение любого атрибута через переменную `$result["attributes"]["название атрибута"]`.

show_last_order module="cart" – выводит информацию о последнем совершенном заказе.

Атрибуты:

- **defer** – маркер отложенной загрузки шаблонного тега: **event** – загрузка контента только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** – загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка контента только по желанию пользователя;
- **defer_title** – текстовая строка, выводимая на месте появления загружаемого контента с помощью отложенной загрузки шаблонного тега;

— **template** – шаблон тега (файл `modules/cart/views/cart.view.show_last_order_template.php`; по умолчанию шаблон `modules/cart/views/cart.view.show_last_order.php`).

Пример:

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

В шаблоне тега можно получить значение любого атрибута через переменную `$result["attributes"][$"название атрибута"]`.

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 артикулу», «Искать товары по акции», «Искать по новинкам», «Искать по хитам»). 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:

- **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. Можно указать отрицательное значение, тогда указанные страницы будут исключены из списка;
- **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 отрицательное значение, тогда указанные категории будут исключены из списка. Можно указать значение **all**, тогда поиск будет осуществляться по всем категориям товаров и в форме будут участвовать только общие характеристики. Атрибут не обязателен;
- **ajax** – подгружать результаты поиска без перезагрузки страницы: **true** – результаты поиска подгружаются, по умолчанию будет перезагружена вся страница. Результаты подгружаются только если открыта страница со списком товаров, иначе поиск работает обычным образом;
- **only_module** – выводить форму поиска только на странице модуля «Магазин»: **true** – выводить форму только на странице модуля, по умолчанию форма будет выводиться на всех страницах;
- **defer** – маркер отложенной загрузки шаблонного тега: **event** – загрузка контента только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** –

загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка контента только по желанию пользователя;

— **defer_title** – текстовая строка, выводимая на месте появления загружаемого контента с помощью отложенной загрузки шаблонного тега;

— **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

В шаблоне тега можно получить значение любого атрибута через переменную `$result["attributes"]["название атрибута"]`.

База данных

[{shop}](#) – Товары

[{shop_additional_cost}](#) – Сопутствующие услуги

[{shop_additional_cost_category_rel}](#) – Связь сопутствующих услуг и категорий

[{shop_additional_cost_rel}](#) – Связь сопутствующих услуг и товаров

[{shop_brand}](#) – Бренды

[{shop_brand_category_rel}](#) – Связи производителей и категорий

[{shop_cart}](#) – Товары в корзине

[{shop_category}](#) – Категории товаров

[{shop_category_parents}](#) – Родительские связи категорий товаров

[{shop_category_rel}](#) – Связи товаров и категорий

[{shop_counter}](#) – Счетчик просмотров товаров

[{shop_currency}](#) – Дополнительные валюты магазина

[{shop_delivery}](#) – Способы доставки

[{shop_delivery_thresholds}](#) – Стоимость способов доставки

[{shop_discount}](#) – Скидки

[{shop_discount_coupon}](#) – Купоны на скидку

[{shop_discount_object}](#) – Товары и категории, на которые действуют скидки

[{shop_discount_person}](#) – Пользователи, для которых действуют скидки

[{shop_files_codes}](#) – Коды для скачивания товаров-нематериальных активов

[{shop_import}](#) – Описание полей файлов импорта

[{shop_import_category}](#) – Описание файлов импорта

[{shop_order}](#) – Заказы

[{shop_order_additional_cost}](#) – Сопутствующие услуги, включенные в заказ

[{shop_order_goods}](#) – Товары в заказе

[{shop_order_goods_param}](#) – Дополнительные характеристики товаров в заказе

[{shop_order_param}](#) – Поля конструктора формы оформления заказа

[{shop_order_param_element}](#) – Значения полей конструктора оформления заказа

[{shop_order_param_select}](#) – Варианты значений полей конструктора оформления заказа типа список

[{shop_order_param_user}](#) – Значения полей конструктора оформления заказа, предзаполненные пользователями

[{shop_order_status}](#) – Статусы заказов

[{shop_param}](#) – Дополнительные характеристики товаров

[{shop_param_category_rel}](#) – Связи дополнительных характеристик товаров и категорий

[{shop_param_element}](#) – Значения дополнительных характеристик товаров

[{shop_param_select}](#) – Варианты значений дополнительных характеристик товаров типа список

[{shop_price}](#) – Цены товаров

[{shop_price_image_rel}](#) – Изображения товаров, прикрепленные к цене

[{shop_price_param}](#) – Дополнительные характеристики, учитываемые в цене

[{shop_rel}](#) – Связи похожих товаров

[{shop_waitlist}](#) – Товары в списке ожидания

[{shop_wishlist}](#) – Товары в списке пожеланий

Файлы

1. *modules/cart/cart.php* – контроллер модуля «Корзина товаров, оформление заказа»;
2. *modules/cart/cart.action.php* – обрабатывает полученные данные из формы;
3. *modules/cart/cart.inc.php* – подключение модуля «Корзина товаров, оформление заказа»;

4. *modules/cart/cart.model.php* – модель модуля «Корзина товаров, оформление заказа»;
5. *modules/cart/js/cart.form.js* – JS-сценарий модуля «Корзина товаров, оформление заказа»;
6. *modules/cart/js/cart.show_block.js* – JS-сценарий блока корзины;
7. *modules/cart/views/cart.view.form.php* – шаблон формы редактирования корзины товаров, оформления заказа;
8. *modules/cart/views/cart.view.images.php* – шаблон прикрепленных изображений;
9. *modules/cart/views/cart.view.info.php* – шаблон информации о товарах в корзине;
10. *modules/cart/views/cart.view.one_click.php* – шаблон форма оформления заказа в один клик;
11. *modules/cart/views/cart.view.payment.php* – шаблон формы платежной системы;
12. *modules/cart/views/cart.view.result.php* – шаблон подтверждения/опровержения платежа;
13. *modules/cart/views/cart.view.show_block.php* – шаблон блока корзины;
14. *modules/cart/views/cart.view.show_last_order.php* – шаблон вывода информации о последнем совершенном заказе;
15. *modules/cart/views/cart.view.table.php* – шаблон таблицы с товарами в корзине;
16. *modules/cart/views/cart.view.table_mail.php* – шаблон таблицы с товарами, отправляемый пользователю по почте;
17. *modules/shop/admin/js/shop.admin.additionalcost.js* – услуги, JS-сценарий;
18. *modules/shop/admin/js/shop.admin.config.js* – настройки модуля, JS-сценарий;
19. *modules/shop/admin/js/shop.admin.discount.js* – редактирование способов доставки, JS-сценарий;
20. *modules/shop/admin/js/shop.admin.importexport.js* – импорт/экспорт данных, JS-сценарий;
21. *modules/shop/admin/js/shop.admin.js* – редактирование товаров, JS-сценарий;
22. *modules/shop/admin/js/shop.admin.order.js* – редактирование заказов, JS-сценарий;
23. *modules/shop/admin/js/shop.admin.orderparam.js* – конструктор формы оформления заказа, JS-сценарий;

- 24. *modules/shop/admin/js/shop.admin.param.js* – редактирование дополнительных характеристик товаров, JS-сценарий;
- 25. *modules/shop/admin/shop.admin.php* – редактирование товаров;
- 26. *modules/shop/admin/shop.admin.action.php* – обработка POST-запросов в административной части модуля;
- 27. *modules/shop/admin/shop.admin.additionalcost.php* – дополнительная стоимость;
- 28. *modules/shop/admin/shop.admin.brand.php* – редактирование производителей;
- 29. *modules/shop/admin/shop.admin.cart.php* – брошенные корзины;
- 30. *modules/shop/admin/shop.admin.category.php* – редактирование категорий магазина;
- 31. *modules/shop/admin/shop.admin.config.php* – настройки модуля;
- 32. *modules/shop/admin/shop.admin.counter.php* – статистика просмотров;
- 33. *modules/shop/admin/shop.admin.currency.php* – валюты;
- 34. *modules/shop/admin/shop.admin.discount.php* – редактирование скидок;
- 35. *modules/shop/admin/shop.admin.import.php* – импорт;
- 36. *modules/shop/admin/shop.admin.importexport.php* – администрирование импорта/экспорт данных;
- 37. *modules/shop/admin/shop.admin.importexport.category.php* – список описанных файлов;
- 38. *modules/shop/admin/shop.admin.importexport.element.php* – импорт/экспорт данных;
- 39. *modules/shop/admin/shop.admin.inc.php* – подключение модуля к административной части других модулей;
- 40. *modules/shop/admin/shop.admin.menu.php* – карта ссылок для модуля «Меню на сайте»;
- 41. *modules/shop/admin/shop.admin.order.php* – редактирование заказов;
- 42. *modules/shop/admin/shop.admin.order.count.php* – количество новых заказов для меню административной панели;
- 43. *modules/shop/admin/shop.admin.order.dashboard.php* – заказы для событий;
- 44. *modules/shop/admin/shop.admin.ordercount.php* – отчет о продажах;
- 45. *modules/shop/admin/shop.admin.orderparam.php* – конструктор формы оформления заказа;

- 46. *modules/shop/admin/shop.admin.orderstatus.php* – статусы заказа;
- 47. *modules/shop/admin/shop.admin.param.php* – редактирование дополнительных характеристик товаров;
- 48. *modules/shop/admin/shop.admin.view.php* – шаблон модуля в административной части;
- 49. *modules/shop/admin/shop.admin.waitlist.php* – список ожиданий;
- 50. *modules/shop/admin/shop.admin.wishlist.php* – список желаний в административной части;
- 51. *modules/shop/inc/shop.inc.order.php* – подключение модуля «Магазин» для работы с заказами;
- 52. *modules/shop/inc/shop.inc.price.php* – подключение модуля «Магазин» для работы с ценами;
- 53. *modules/shop/js/shop.buy_form.js* – JS-сценарий модуля;
- 54. *modules/shop/js/shop.compare.js* – JS-сценарий сравнения товаров;
- 55. *modules/shop/js/shop.id.js* – JS-сценарий модуля;
- 56. *modules/shop/js/shop.show_search.js* – JS-сценарий формы поиска по товарам;
- 57. *modules/shop/shop.php* – контроллер модуля «Магазин»;
- 58. *modules/shop/shop.1c.php* – интеграция с системой 1С:Предприятие;
- 59. *modules/shop/shop.action.php* – обработка запроса при добавлении товара в корзину;
- 60. *modules/shop/shop.export.php* – экспорт товаров;
- 61. *modules/shop/shop.google.php* – выгрузка в систему Google Merchant;
- 62. *modules/shop/shop.inc.php* – подключение модуля «Магазин»;
- 63. *modules/shop/shop.install.php* – установка модуля;
- 64. *modules/shop/shop.model.php* – модель модуля «Магазин»;
- 65. *modules/shop/shop.search.php* – настройки для поисковой индексации для модуля «Поиск»;
- 66. *modules/shop/shop.sitemap.php* – карта ссылок для модуля «Карта сайта»;
- 67. *modules/shop/shop.yandex.php* – выгрузка в систему Яндекс.Маркет;

- 68. *modules/shop/views/m/shop.view.id.php* – шаблон страницы товара;
- 69. *modules/shop/views/shop.view.buy_form.php* – шаблон кнопки «Купить», в котором характеристики, влияющие на цену выводятся в виде выпадающего списка;
- 70. *modules/shop/views/shop.view.buy_form_list.php* – шаблон кнопки «Купить», в котором характеристики, влияющие на цену выводятся в виде выпадающего списка;
- 71. *modules/shop/views/shop.view.buy_form_order_rel.php* – шаблон кнопки «Купить» для блока товаров;
- 72. *modules/shop/views/shop.view.compare.php* – шаблон страницы сравнения товаров;
- 73. *modules/shop/views/shop.view.compared_goods_list.php* – шаблон кнопки «Сравнить выбранные товары»;
- 74. *modules/shop/views/shop.view.compare_form.php* – шаблон кнопки «Сравнить» для товаров;
- 75. *modules/shop/views/shop.view.compare_param.php* – шаблон дополнительных характеристик товара на странице сравнения;
- 76. *modules/shop/views/shop.view.first_page.php* – шаблон первой страницы модуля, если в настройках модуля подключен параметр «Использовать категории»;
- 77. *modules/shop/views/shop.view.id.php* – шаблон страницы товара;
- 78. *modules/shop/views/shop.view.list.php* – шаблон списка товаров;
- 79. *modules/shop/views/shop.view.list_search.php* – шаблон списка товаров для поиска;
- 80. *modules/shop/views/shop.view.param.php* – шаблон дополнительных характеристик товара;
- 81. *modules/shop/views/shop.view.rows.php* – шаблон элементов в списке товаров;
- 82. *modules/shop/views/shop.view.show_add_coupon.php* – шаблон формы активации купона;
- 83. *modules/shop/views/shop.view.show_block.php* – шаблон блока товаров;
- 84. *modules/shop/views/shop.view.show_block_left.php* – шаблон блока товаров;
- 85. *modules/shop/views/shop.view.show_block_order_rel.php* – шаблон блока товаров, которые обычно покупают с текущим товаром;
- 86. *modules/shop/views/shop.view.show_block_rel.php* – шаблон блока похожих товаров;
- 87. *modules/shop/views/shop.view.show_brand.php* – шаблон блока производителей;

- 88. *modules/shop/views/shop.view.show_category.php* – шаблон блока категорий;
- 89. *modules/shop/views/shop.view.show_category_level.php* – шаблон вложенных уровней блока категорий;
- 90. *modules/shop/views/shop.view.show_search.php* – шаблон форма поиска по товарам;
- 91. *modules/shop/views/shop.view.sort_block.php* – шаблон блока «Сортировать» с ссылками на направление сортировки;
- 92. *modules/wishlist/js/wishlist.form.js* – JS-сценарий модуля «Список желаний»;
- 93. *modules/wishlist/views/wishlist.view.form.php* – шаблон формы редактирования списка желаний;
- 94. *modules/wishlist/views/wishlist.view.info.php* – шаблон информации о товарах в списке пожеланий;
- 95. *modules/wishlist/views/wishlist.view.show_block.php* – шаблон блока списка желаний;
- 96. *modules/wishlist/views/wishlist.view.table.php* – шаблон таблицы с товарами в списке желаний;
- 97. *modules/wishlist/wishlist.php* – контроллер модуля «Список пожеланий»;
- 98. *modules/wishlist/wishlist.action.php* – обработка запроса при пересчете суммы покупки в списке желаний;
- 99. *modules/wishlist/wishlist.inc.php* – подключение модуля «Список пожеланий»;
- 100. *modules/wishlist/wishlist.model.php* – модель модуля Список желаний.

Фотогалерея

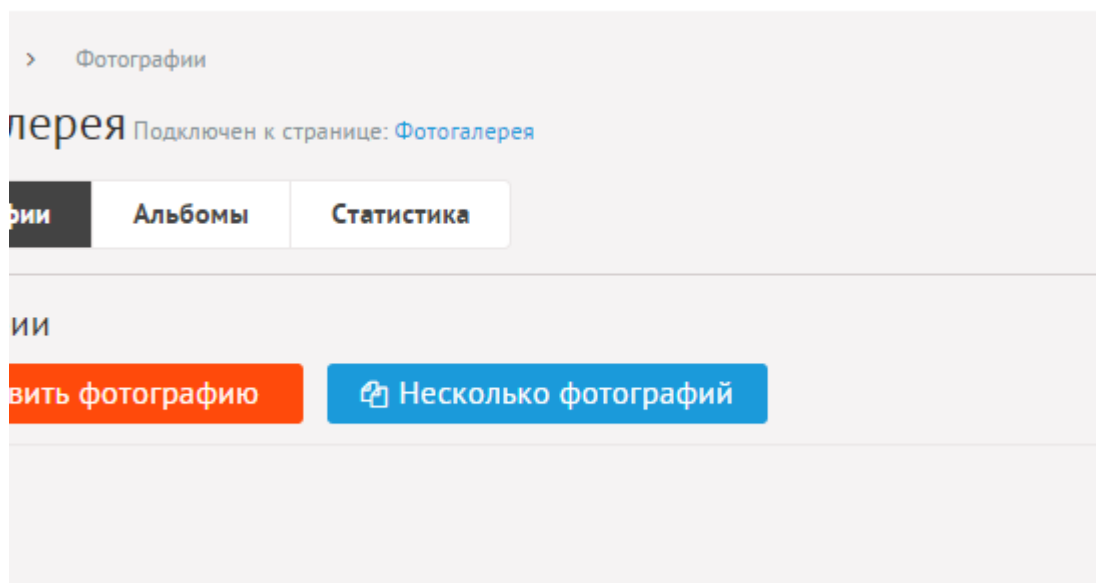
- [Фотографии](#)
- [Альбомы](#)
- [Статистика](#)
- [Настройки](#)
- [Шаблонные теги](#)
- [База данных](#)
- [Файлы](#)

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

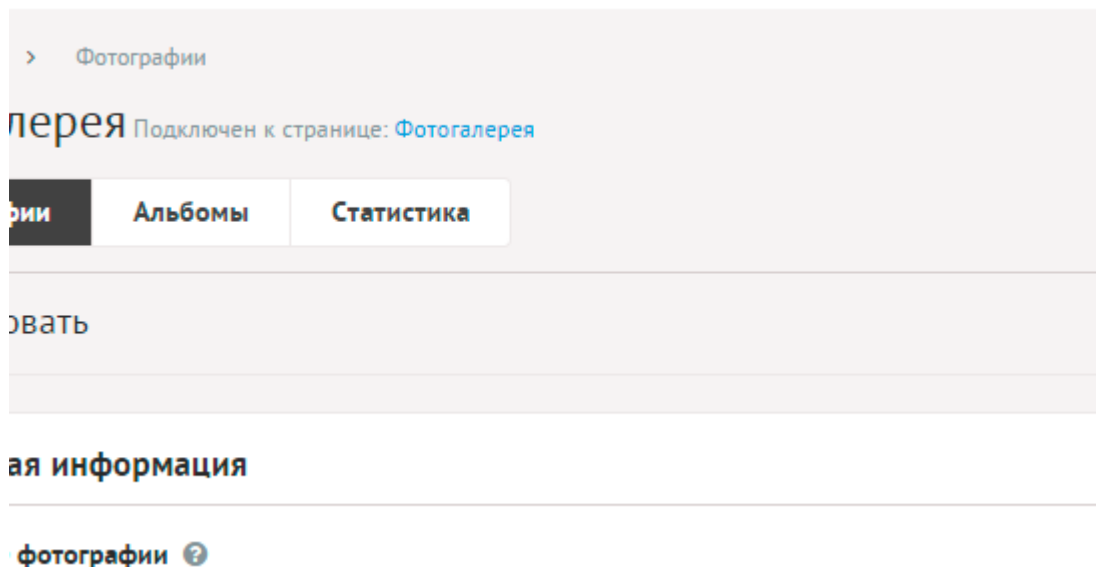
Модуль состоит из двух частей: альбомы и фотографии.

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).



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.
- **Альбом** – Альбом, к которому относится фотография. Список альбомов редактируется во вкладке выше. 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».
- **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»](#). Опция «HTML-код» позволяет отключить визуальный редактор для текущего поля. Значение этой настройки будет учитываться и при последующем редактировании. 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»](#). Опция «HTML-код» позволяет отключить визуальный редактор для текущего поля. Значение этой настройки будет учитываться и при последующем редактировании. Параметр имеет разные значения для разных языковых версий сайта.
- **Динамические блоки.**
- **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.

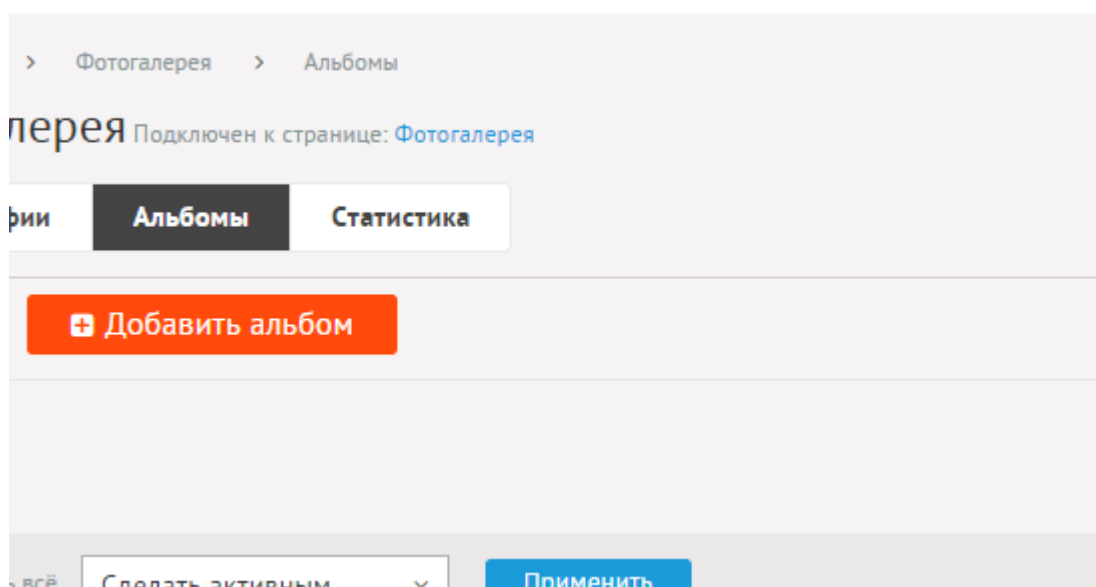
- **Теги** – Добавление тегов к фотографии. Можно добавить либо новый тег, либо открыть и выбрать из уже существующих тегов. Параметр выводится, если в настройках модуля включен параметр «Подключить теги».
- **Счетчик просмотров** – Количество просмотров на сайте текущей фотографии. Статистика ведется 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».

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*.
- **Раздел сайта** – Перенос фотографии на другую страницу сайта, к которой прикреплен модуль. Параметр выводится, если в настройках модуля отключена опция «Использовать альбомы», если опция подключена, то раздел сайта задается такой же, как у основного альбома.
- **Параметры SEO.**
- **Заголовок окна в браузере, тег Title** – Если не заполнен, тег *Title* будет автоматически сформирован как «Название фотографии – 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, 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.
- **Канонический тег** – URL канонической страницы вида: <http://site.ru/pseudossylka/>, на которую переносится "ссылочный вес" данной страницы. Используется для страниц с похожим или дублирующимся контентом (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).

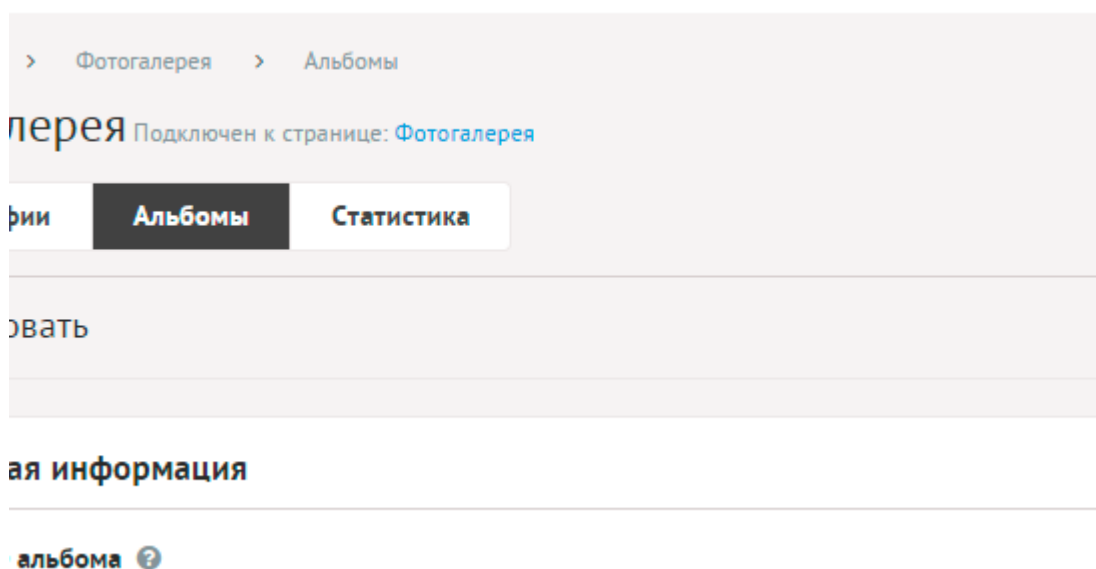
- **Редирект на текущую страницу со страницы** – Позволяет делать редирект с указанной страницы на текущую.
- **Не индексировать** – Запрет индексации текущей страницы, если отметить, у страницы выведется тег: `<meta name="robots" content="noindex">` (SEO-специалисту).
- **Changefreq** – Вероятная частота 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).
- **Параметры показа.**
- **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).
- **Оформление.**
- **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!).
- **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*.

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.

При удалении альбома удаляются все вложенные альбомы и фотографии.



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.
- **Изображения** – Возможность загрузки нескольких изображений. Варианты размера

загружаемых изображений определяются в настройках. Параметр выводится, если в настройках модуля отмечена опция «Использовать изображения для альбомов».

- **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»](#). Опция «HTML-код» позволяет отключить визуальный редактор для текущего поля. Значение этой настройки будет учитываться и при последующем редактировании. 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»](#). Опция «HTML-код» позволяет отключить визуальный редактор для текущего поля. Значение этой настройки будет учитываться и при последующем редактировании. Параметр имеет разные значения для разных языковых версий сайта.
- **Динамические блоки.**
- **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».
- **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:

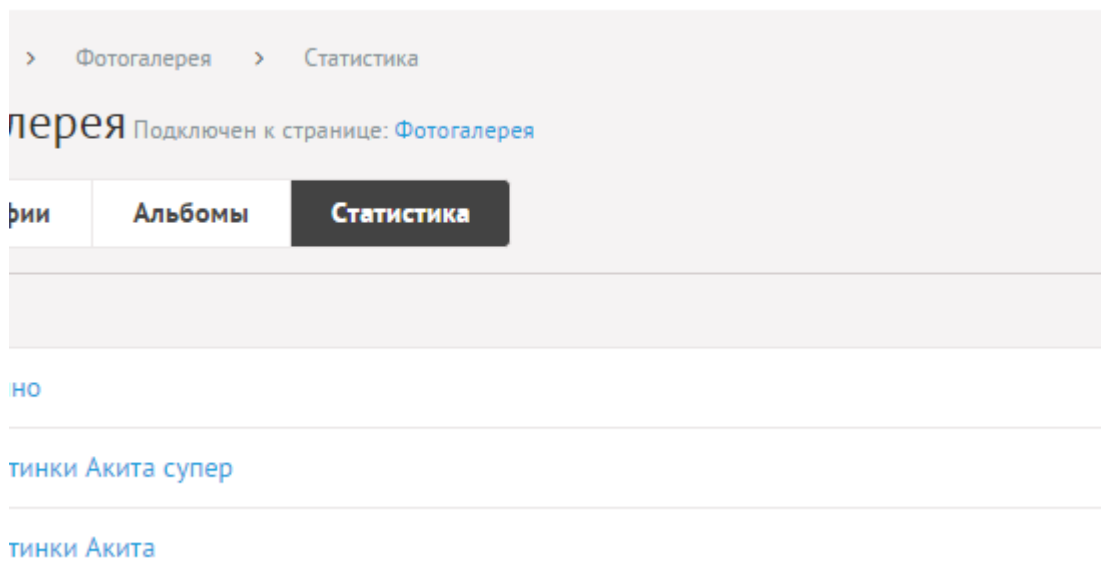
- **Номер** – Номер элемента в БД (веб-мастеру и программисту).
- **Редактор** – Изменяется после первого сохранения. Показывает, кто из администраторов сайта первый правил текущую страницу.
- **Время последнего изменения** – Изменяется после сохранения элемента. Отдается в заголовке *Last Modify*.
- **Раздел сайта** – Перенос альбома на другую страницу сайта, к которой прикреплен модуль (администратору сайта).
- **Параметры SEO.**
- **Заголовок окна в браузере, тег Title** – Если не заполнен, тег *Title* будет автоматически сформирован как «Название категории – 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, 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.

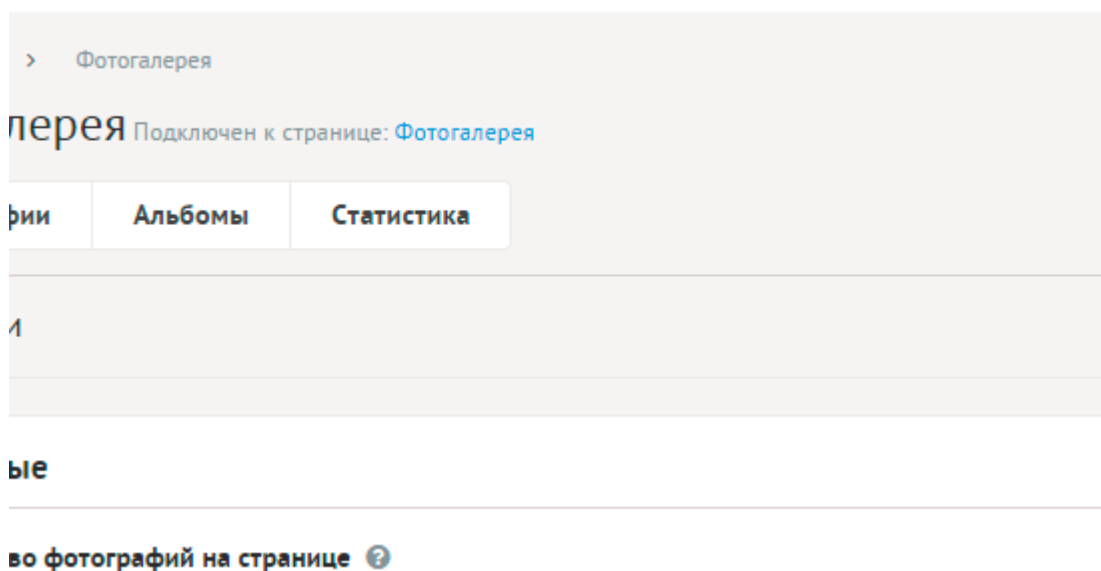
- **Канонический тег** – URL канонической страницы вида: <http://site.ru/pseudossylka/>, на которую переносится "ссылочный вес" данной страницы. Используется для страниц с похожим или дублирующимся контентом (SEO-специалисту). Параметр имеет разные значения для разных языковых версий сайта.
- **Псевдоссылка** – 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.
- **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).
- **Параметры показа.**
- **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).
- **Оформление.**
- **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 списка элементов** – Шаблон вывода контента модуля на странице элементов списка в категории (веб-мастеру и программисту, не меняйте этот параметр, если не уверены в результате!). Значение параметра важно для AJAX.
- **Шаблон страницы элемента** – 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!).

Статистика



Настройки



- **Основные.**
- **Количество фотографий на странице** – Количество одновременно выводимых фотографий в списке.
- **Количество альбомов на странице** – Количество одновременно выводимых альбомов в списке на первой страницы модуля. Параметр появляется, если отмечена опция «Использовать альбомы».
- **Включить «Показать ещё»** – На странице фотографий появится кнопка «Показать ещё». Увеличивает количество одновременно выводимых фотографий в списке.
- **Активировать фотографии после групповой загрузки** – 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.
- **Альбомы.**
- **Использовать альбомы** – Разделение фотогалереи на альбомы.
- **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 вложенного альбома** – Для первой страницы модуля и для страницы альбома. 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 альбому.
- **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».

- **Подключения.**
- **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 альбомам** – 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».
- **Автогенерация для SEO.**
- **Шаблон для автоматического генерирования Title** – Если шаблон задан и для фотографии не прописан заголовок *Title*, то заголовок автоматически генерируется по шаблону. В шаблон можно добавить:
 - %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 и для альбома не прописан заголовок *Title*, то заголовок автоматически генерируется по шаблону. В шаблон можно добавить:
 - %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 и для фотографии не заполнено поле *Keywords*, то поле *Keywords* автоматически

генерируется по шаблону. В шаблон можно добавить:

- %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 альбома** – Если шаблон задан и для альбома не заполнено поле *Keywords*, то поле *Keywords* автоматически генерируется по шаблону. В шаблон можно добавить:

- %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 и для фотографии не заполнено поле *Description*, то поле *Description* автоматически генерируется по шаблону. В шаблон можно добавить:

- %name – название,
- %category – название альбома,
- %parent_category – название альбома верхнего уровня,
- %anons – краткое описание (SEO-специалисту).

Параметр имеет разные значения для разных языковых версий сайта.

- **Шаблон для автоматического генерирования Description для альбома** – Если шаблон задан и для альбома не заполнено поле *Description*, то поле *Description* автоматически генерируется по шаблону. В шаблон можно добавить:

- %name – название альбома,
- %parent – название альбома верхнего уровня,
- %anons – краткое описание (SEO-специалисту).

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

- **Оформление.**
- **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 элементов в списке** – По умолчанию *modules/photo/views/photo.view.rows.php*. Параметр для разработчиков! Не устанавливайте, если не уверены в результате. Значение параметра важно для AJAX.
- **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 элементов в списке первой страницы модуля (если подключены категории)** – По умолчанию *modules/photo/views/photo.view.fitst_page.php*. Параметр для разработчиков! Не устанавливайте, если не уверены в результате. Значение параметра важно для AJAX.
- **Шаблон для страницы элемента** – По умолчанию, *modules/photo/views/photo.view.id.php*. Параметр для разработчиков! Не

устанавливайте, если не уверены в результате.

- **Дополнительно.**
- **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. Можно указать отрицательное значение, тогда будут исключены фотографии из указанного раздела. По умолчанию выбираются все страницы;
- **cat_id** – альбомы фотографий, если в настройках модуля отмечено «Использовать альбомы». Идентификаторы альбомов перечисляются через запятую. Можно указать отрицательное значение, тогда будут исключены фотографии из указанной категории. Можно указать значение **current**, тогда будут показаны фотографии из текущей (открытой) категории или из всех категорий, если ни одна категория не открыта. 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;
- **only_module** – выводить блок только на странице, к которой прикреплен модуль «Фотогалерея»: **true** – выводить блок только на странице модуля, по умолчанию блок будет выводиться на всех страницах;
- **tag** – тег, прикрепленный к фотографиям;
- **defer** – маркер отложенной загрузки шаблонного тега: **event** – загрузка контента только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** – загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка контента только по желанию пользователя;

- **defer_title** – текстовая строка, выводимая на месте появления загружаемого контента с помощью отложенной загрузки шаблонного тега;
- **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

В шаблоне тега можно получить значение любого атрибута через переменную `$result["attributes"]["название атрибута"]`.

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;
- **defer** – маркер отложенной загрузки шаблонного тега: **event** – загрузка контента только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** – загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка контента только по желанию пользователя;
- **defer_title** – текстовая строка, выводимая на месте появления загружаемого контента с помощью отложенной загрузки шаблонного тега;
- **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

В шаблоне тега можно получить значение любого атрибута через переменную `$result["attributes"]["название атрибута"]`.

База данных

[{photo}](#) – Фотографии

[{photo_category}](#) – Альбомы фотографий

[{photo_category_parents}](#) – Родительские связи альбомов фотографий

[{photo_category_rel}](#) – Связи фотографий с альбомами

[{photo_counter}](#) – Счетчик просмотров фотографий

[{photo_rel}](#) – Связи похожих фотографий

Файлы

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

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

модуля «Теги»;

17. *modules/photo/views/photo.view.rows.php* – шаблон элементов в списке фотографий;

18. *modules/photo/views/photo.view.rows_block.php* – шаблон блока фотографий;

19. *modules/photo/views/photo.view.rows_block_rel.php* – шаблон блока похожих фотографий;

20. *modules/photo/views/photo.view.rows_tags.php* – шаблон элементов в списке фотографий для модуля «Теги»;

21. *modules/photo/views/photo.view.show_block.php* – шаблон блока фотографий;

22. *modules/photo/views/photo.view.show_block_rel.php* – шаблон блока похожих фотографий.

Статьи

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

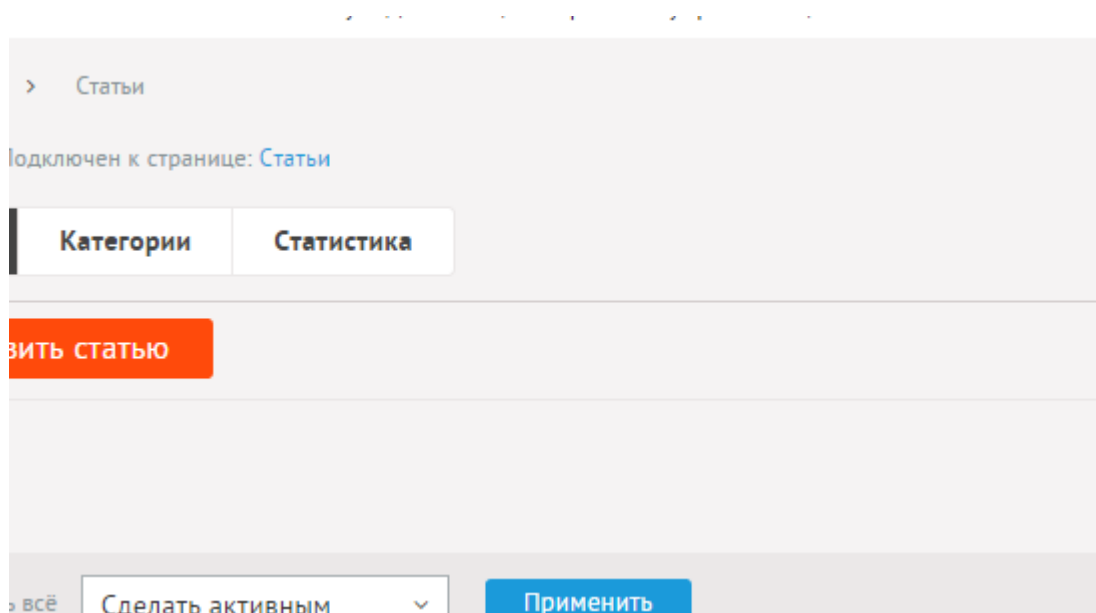
Модуль «Статьи» отображает статьи.

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

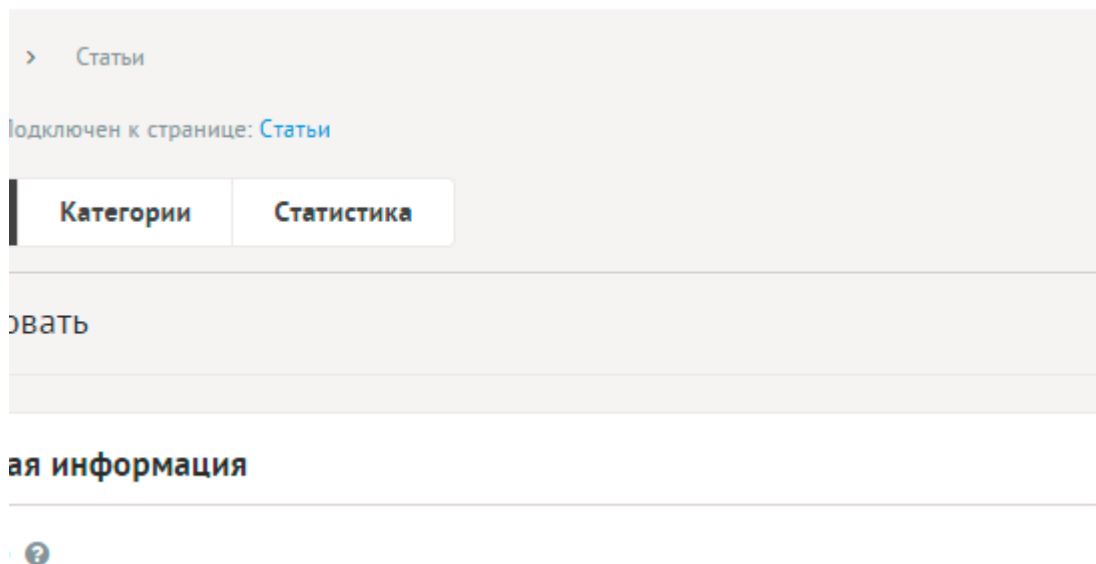
Модуль «Статьи» можно прикреплять к различным страницам сайта. Если модуль прикреплен к нескольким страницам, в административной части модуля появляется специальный фильтр «Раздел сайта» с возможностью фильтрации вывода (списка статей, категорий и настроек) по принадлежности к странице.

Модуль состоит из двух частей: категории статей и статьи.

Статьи



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



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.
- **Категория** – Категория, к которой относится статья. Список категорий редактируется во вкладке выше. 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».
- **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»](#). Опция «HTML-код» позволяет отключить визуальный редактор для текущего поля. Значение этой настройки будет учитываться и при последующем редактировании. 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»](#). Опция «HTML-код» позволяет отключить визуальный редактор для текущего поля. Значение этой настройки будет

учитываться и при последующем редактировании. Параметр имеет разные значения для разных языковых версий сайта.

- **Динамические блоки.**
- **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».
- **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.
- **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».

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*.
- **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.
- **Параметры SEO.**
- **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.
- **Канонический тег** – URL канонической страницы вида: <http://site.ru/pseudossylka/>, на которую переносится "ссылочный вес" данной страницы. Используется для страниц с похожим или дублирующимся контентом (SEO-специалисту). Параметр имеет разные

значения для разных языковых версий сайта.

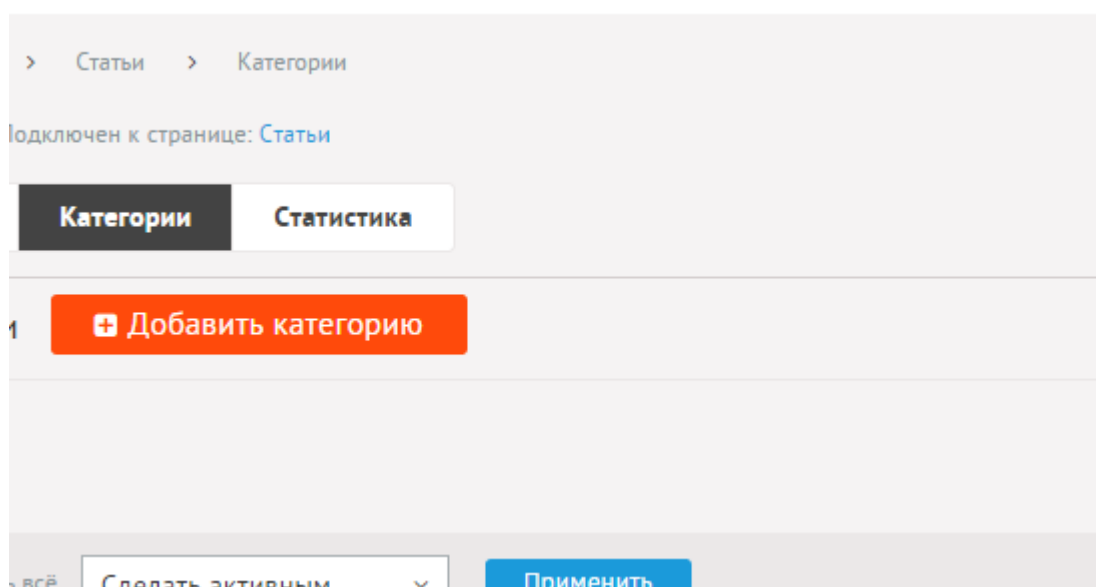
- **Псевдоссылка** – ЧПУ, т.е. адрес страницы вида: <http://site.ru/pseudossylka/>. Смотрите параметры сайта (SEO-специалисту).
- **Редирект на текущую страницу со страницы** – Позволяет делать редирект с указанной страницы на текущую.
- **Не индексировать** – Запрет индексации текущей страницы, если 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).
- **Параметры показа.**
- **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).
- **Оформление.**
- **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!).
- **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*.

В модуль «Статьи» встроена автоматическая публикация статей в формате RSS.

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

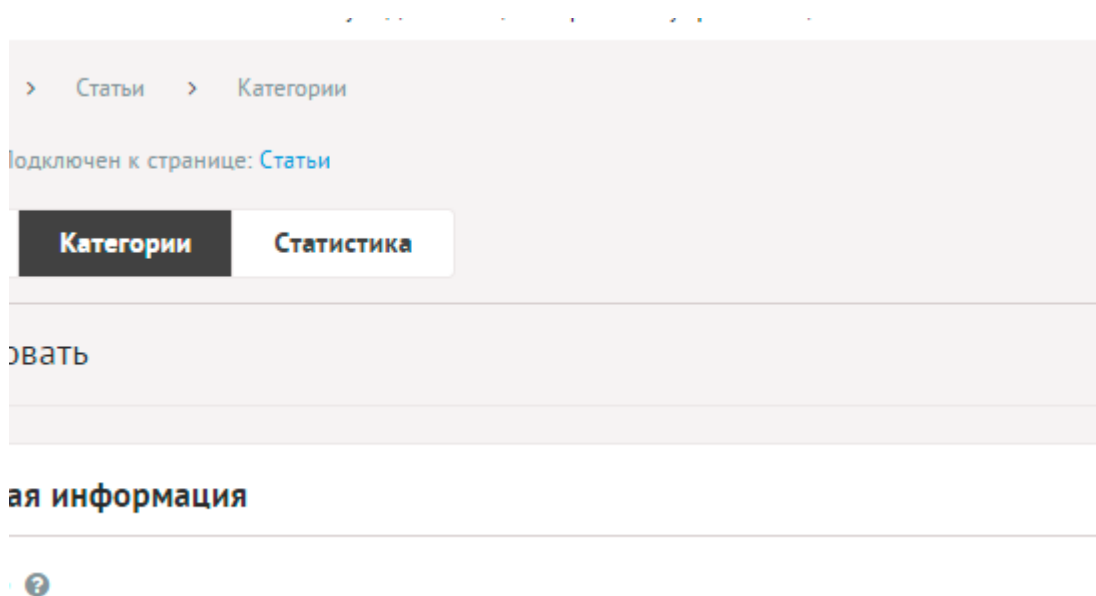
У англоязычной версии сайта адрес RSS-ленты новостей будет иметь вид:
<http://site.ru/eng/clauses/rss/>

Категории



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

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».

- **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»](#). Опция «HTML-код» позволяет отключить визуальный редактор для текущего поля. Значение этой настройки будет учитываться и при последующем редактировании. 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»](#). Опция «HTML-код» позволяет отключить визуальный редактор для текущего поля. Значение этой настройки будет учитываться и при последующем редактировании. Параметр имеет разные значения для разных языковых версий сайта.
- **Динамические блоки.**
- **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».
- **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:

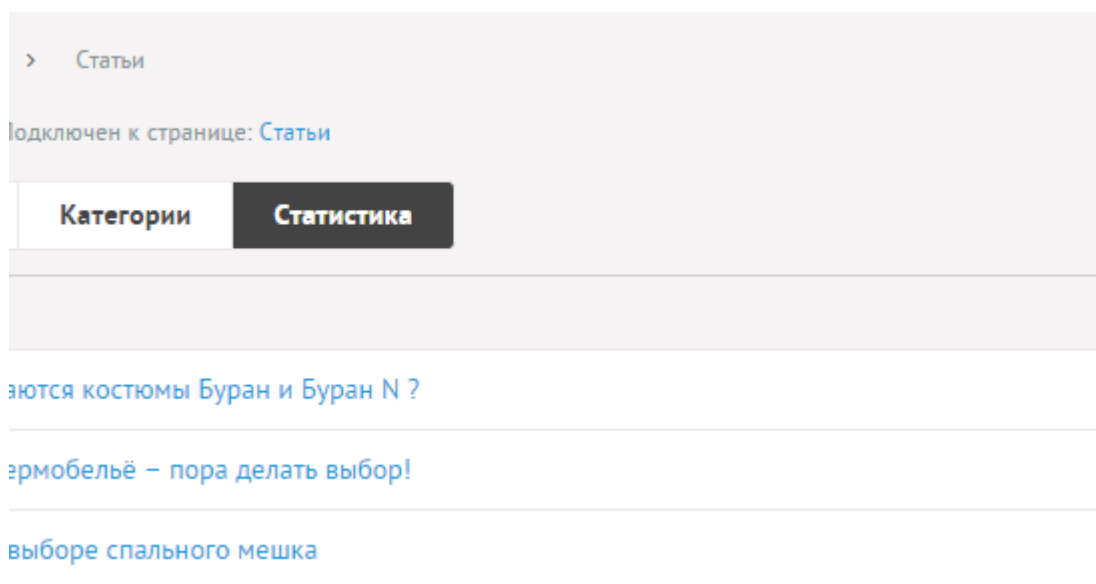
- **Номер** – Номер элемента в БД (веб-мастеру и программисту).
- **Редактор** – Изменяется после первого сохранения. Показывает, кто из администраторов сайта первый правил текущую страницу.
- **Время последнего изменения** – Изменяется после сохранения элемента. Отдается в заголовке *Last Modify*.
- **Раздел сайта** – Перенос категории на другую страницу сайта, к которой прикреплен модуль (администратору сайта).
- **Параметры SEO.**
- **Заголовок окна в браузере, тег Title** – Если не заполнен, тег *Title* будет автоматически сформирован как «Название категории – 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, 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.

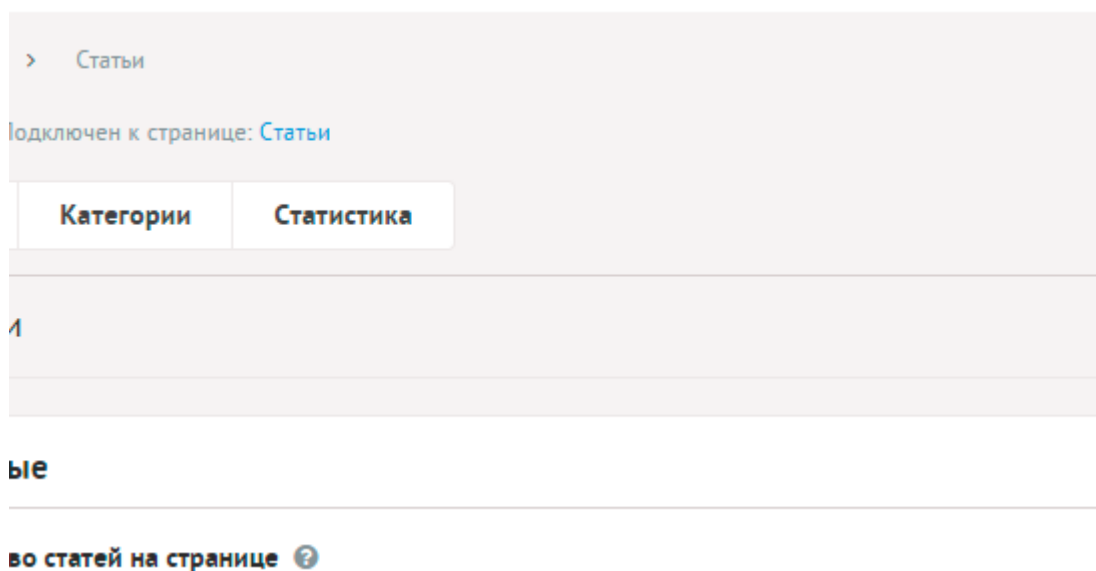
- **Канонический тег** – URL канонической страницы вида: <http://site.ru/pseudossylka/>, на которую переносится "ссылочный вес" данной страницы. Используется для страниц с похожим или дублирующимся контентом (SEO-специалисту). Параметр имеет разные значения для разных языковых версий сайта.
- **Псевдоссылка** – 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.
- **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).
- **Параметры показа.**
- **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).
- **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).
- **Оформление.**
- **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 списка элементов** – Шаблон вывода контента модуля на странице элементов списка в категории (веб-мастеру и программисту, не меняйте этот параметр, если не уверены в результате!). Значение параметра важно для AJAX.
- **Шаблон страницы элемента** – 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!).

Статистика



Настройки



- **Основные.**
- **Количество статей на странице** – Количество одновременно выводимых статей в списке.
- **Включить «Показать ещё»** – На странице статей появится кнопка «Показать ещё». Увеличивает количество одновременно выводимых статей в списке.
- **Формат даты** – Позволяет настроить отображение даты в модуле. Возможные значения:
 - 01.05.2016,
 - 01.05.2016 14:45,
 - 1 мая 2016 г.,
 - 1 мая,

- 1 мая 2016, понедельник,
- вчера 15:30,
- не отображать.
- **Категории.**
- **Выводить на первой странице весь список статей** – По умолчанию на первой странице выводится список категорий с несколькими статьями. Параметр появляется, если отмечена опция «Использовать категории».
- **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 категорий на странице** – Количество одновременно выводимых категорий в списке на первой страницы 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».
- **Изображения.**
- **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».
- **Подключения.**
- **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».
- **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.
- **Автогенерация для SEO.**
- **Шаблон для автоматического генерирования Title** – Если шаблон задан и для статьи не прописан заголовок *Title*, то заголовок автоматически генерируется по шаблону. В шаблон можно добавить:
 - %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 и для категории не прописан заголовок *Title*, то заголовок автоматически

генерируется по шаблону. В шаблон можно добавить:

- %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 и для статьи не заполнено поле *Keywords*, то поле *Keywords* автоматически генерируется по шаблону. В шаблон можно добавить:

- %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 и для категории не заполнено поле *Keywords*, то поле *Keywords* автоматически генерируется по шаблону. В шаблон можно добавить:

- %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 и для статьи не заполнено поле *Description*, то поле *Description* автоматически генерируется по шаблону. В шаблон можно добавить:

- %name – название,
- %category – название категории,
- %parent_category – название категории верхнего уровня,
- %anons – краткое описание (SEO-специалисту).

Параметр имеет разные значения для разных языковых версий сайта.

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

- %name – название категории,
- %parent – название категории верхнего уровня,
- %anons – краткое описание (SEO-специалисту).

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

- **Оформление.**

- **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 элементов в списке** – По умолчанию *modules/courses/views/courses.view.rows.php*. Параметр для разработчиков! Не

- устанавливайте, если не уверены в результате. Значение параметра важно для AJAX.
- **Шаблон для первой страницы модуля (если подключены категории)** – По умолчанию *modules/clauses/views/clauses.view.ftst_page.php*. Параметр для разработчиков! Не устанавливайте, если не уверены в результате.
 - **Шаблон для элементов в списке первой страницы модуля (если подключены категории)** – По умолчанию *modules/clauses/views/clauses.view.ftst_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.
 - **Дополнительно.**
 - **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. Можно указать отрицательное значение, тогда будут исключены статьи из указанного раздела. По умолчанию выбираются все страницы;
- **cat_id** – категории статей, если в настройках модуля отмечено «Использовать категории». Идентификаторы категорий перечисляются через запятую. Можно указать отрицательное значение, тогда будут исключены статьи из указанной категории. Можно указать значение **current**, тогда будут показаны статьи из текущей (открытой) категории или из всех категорий, если ни одна категория не открыта. По умолчанию категория не учитывается, выводятся все статьи;
- **sort** – сортировка статей: по умолчанию как на странице модуля, **date** – по дате, **rand** – в случайном порядке, **keywords** – статьи, похожие по названию для текущей страницы (должен быть подключен модуль «Поиск по сайту» и проиндексированы статьи);
- **images** – количество изображений, прикрепленных к статье;
- **images_variation** – тег размера изображений, задается в настройках модуля;

- **only_module** – выводить блок только на странице, к которой прикреплен модуль «Статьи»: **true** – выводить блок только на странице модуля, по умолчанию блок будет выводиться на всех страницах;
- **tag** – тег, прикрепленный к статьям;
- **defer** – маркер отложенной загрузки шаблонного тега: **event** – загрузка контента только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** – загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка контента только по желанию пользователя;
- **defer_title** – текстовая строка, выводимая на месте появления загружаемого контента с помощью отложенной загрузки шаблонного тега;
- **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

В шаблоне тега можно получить значение любого атрибута через переменную `$result["attributes"]["название атрибута"]`.

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;

— **defer** – маркер отложенной загрузки шаблонного тега: **event** – загрузка контента только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** – загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка контента только по желанию пользователя;

— **defer_title** – текстовая строка, выводимая на месте появления загружаемого контента с помощью отложенной загрузки шаблонного тега;

— **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

В шаблоне тега можно получить значение любого атрибута через переменную `$result["attributes"][$"название атрибута"]`.

База данных

[{clauses}](#) – Статьи

[{clauses_category}](#) – Категории статей

[{clauses_category_parents}](#) – Родительские связи категорий статей

[{clauses_category_rel}](#) – Связи статей и категорий

[{clauses_counter}](#) – Счетчик просмотров статей

[{clauses_rel}](#) – Связи похожих статей

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.rows.php* – шаблон элементов в списке статей;
16. *modules/clauses/views/clauses.view.rows_block.php* – шаблон блока статей;
17. *modules/clauses/views/clauses.view.rows_block_rel.php* – шаблон блока похожих статей;
18. *modules/clauses/views/clauses.view.show_block.php* – шаблон блока статей;
19. *modules/clauses/views/clauses.view.show_block_rel.php* – шаблон блока похожих статей.

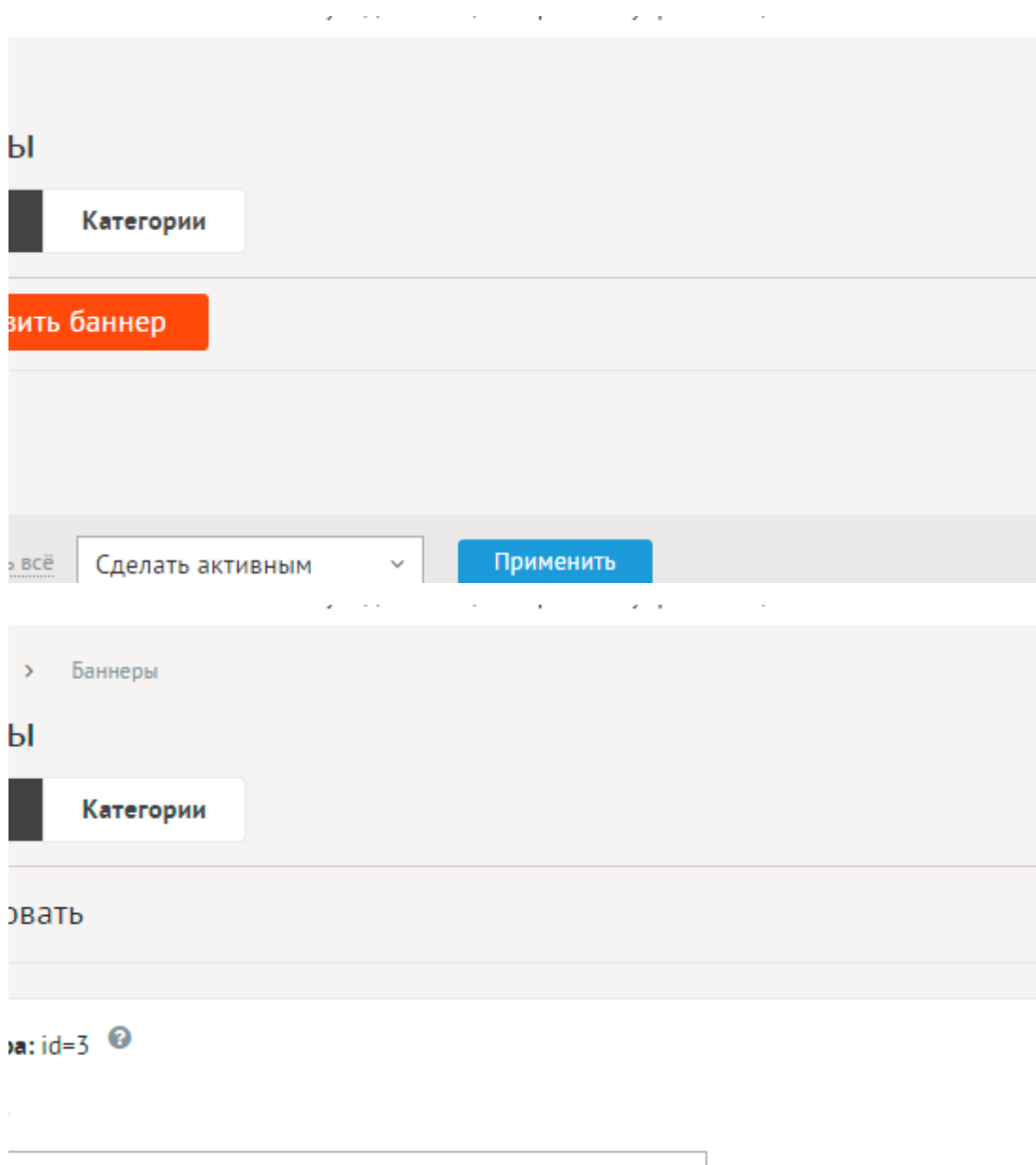
Баннеры

- [Баннеры](#)
- [Категории](#)
- [Настройки](#)
- [Шаблонные теги](#)
- [База данных](#)
- [Файлы](#)

Модуль «Баннеры» отображает баннеры, размещенные на сайте.

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

Баннеры

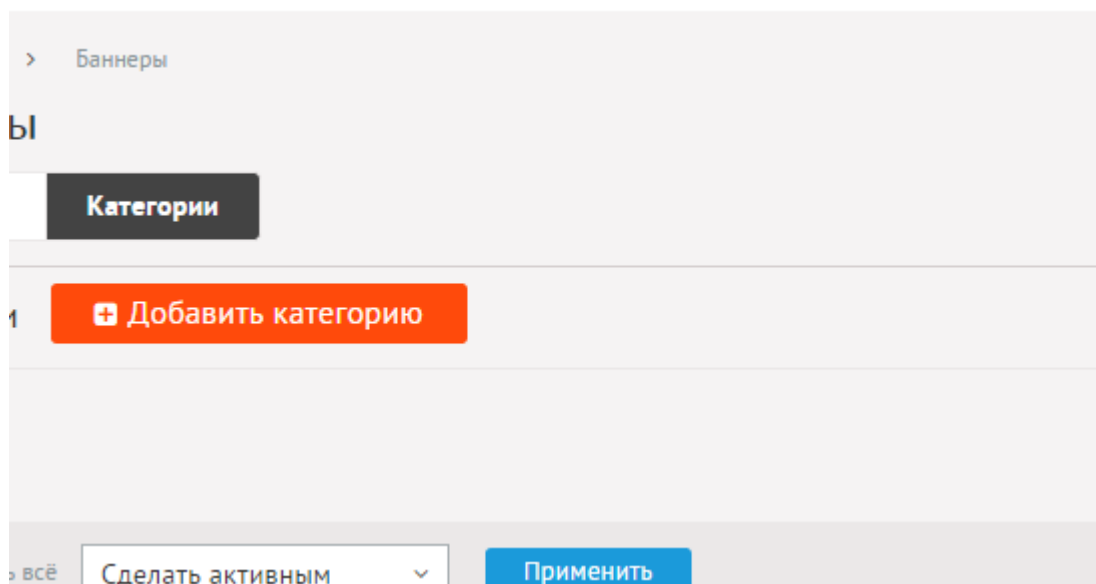


Баннеры имеют следующие характеристики.

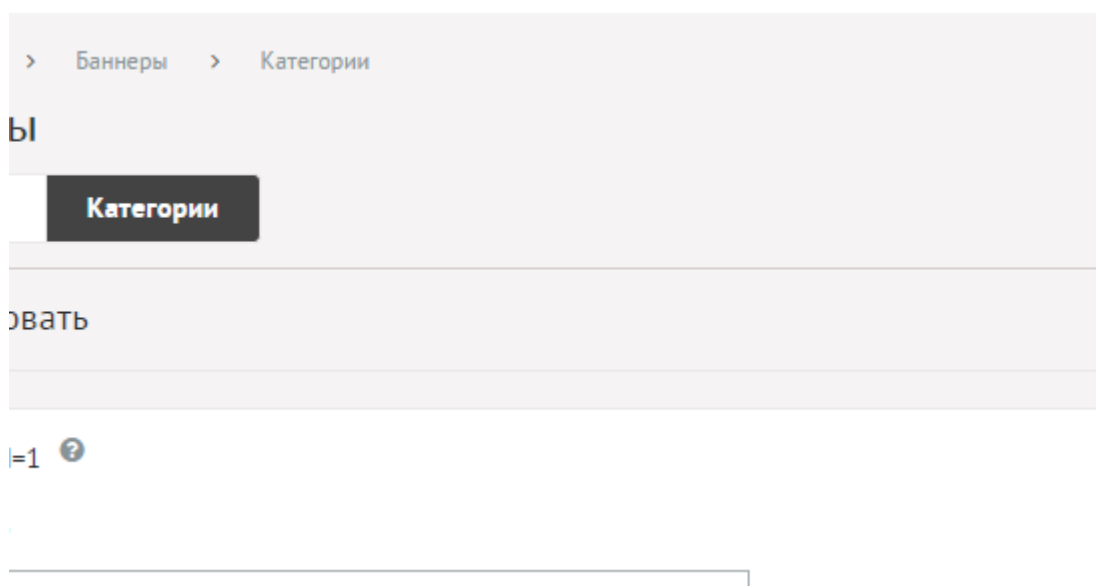
Основные:

- **ID баннера** – Номер элемента в БД (веб-мастеру и программисту).
- **Опубликовать на сайте** – Параметр имеет разные значения для разных языковых версий сайта.
- **Название** – Параметр имеет разные значения для разных языковых версий сайта.
- **Вид баннера** – Изображение, флэш, HTML.
- **Ссылка** – В полном формате *http://www.site.ru/*. Параметр имеет разные значения для разных языковых версий сайта.
- **Открывать в новом окне** – Ссылка на баннер будет открываться в новом окне
- **Описание** – Параметр имеет разные значения для разных языковых версий сайта.
- **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.
- **Всего показов** – Статистика прошедших показов баннера.
- **Всего кликов** – Статистика прошедших кликов по баннеру.
- **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).
- **Ограничить количество показов по кликам** – Ограничение показа до заданного количества.
- **Осталось кликов** – Укажите число, обозначающее, через какое количество кликов скрыть отображение баннера. С каждым кликом цифра в этом поле будет уменьшаться, пока не станет 0 (или пустое поле).
- **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.
- **Категория.**
- **Сортировка: установить перед** – Редактирование порядка следования баннера в списке. Поле доступно для редактирования только для баннеров, отображаемых на сайте.

Категории



Баннеры могут иметь неограниченное количество категорий. При необходимости категории можно не использовать. Для этого нужно отключить параметр «Использовать категории» в конфигурации модуля.



Категории имеют следующие характеристики.

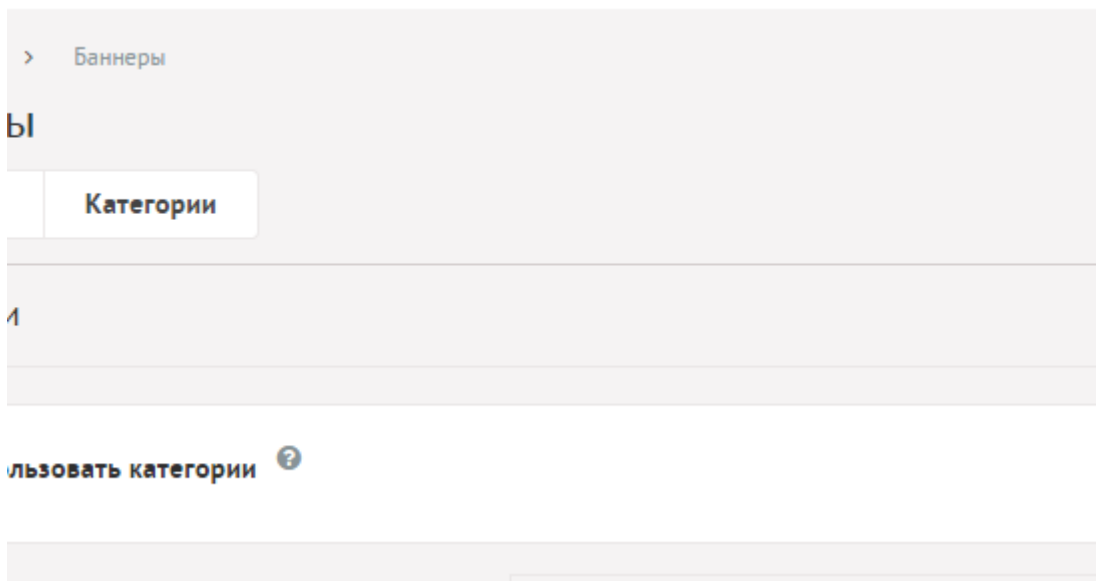
Основные:

- **Номер** – Номер элемента в БД (веб-мастеру и программисту).
- **Название.**
- **Опубликовать на сайте.**
- **Сортировка: установить перед** – Редактирование порядка следования категории в списке. Поле доступно для редактирования только для категорий, отображаемых на сайте.

Настройки

В настройках модуля вы можете выбрать параметр «Использовать категории».

- **Использовать категории** – Позволяет включить/отключить категории баннеров.



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: по умолчанию как в панели администрирования, **date** – по дате, **rand** – в случайном порядке;
- **cat_id** – категория баннеров, если в настройках модуля отмечено «Использовать категории»;
- **defer** – маркер отложенной загрузки шаблонного тега: **event** – загрузка контента только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** – загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка контента только по желанию пользователя;
- **defer_title** – текстовая строка, выводимая на месте появления загружаемого контента с помощью отложенной загрузки шаблонного тега;
- **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" sort="rand">
```

выведет 1 случайный баннер

База данных

[{bs}](#) – Баннеры

[{bs_category}](#) – Категории баннеров

[{bs_site_rel}](#) – Данные о том, на каких страницах сайта выводятся баннеры

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/admin/js/bs.admin.edit.js* – редактирование баннеров, JS-сценарий;
5. *modules/bs/bs.php* – контроллер;
6. *modules/bs/bs.action.php* – обработка запроса при клике на ссылку баннера;
7. *modules/bs/bs.install.php* – установка модуля;
8. *modules/bs/bs.model.php* – модель;
9. *modules/bs/js/bs.show_block.js* – JS-сценарий обработки клика по баннеру;
10. *modules/bs/views/bs.view.show_block.php* – шаблон блока баннеров;
11. *modules/bs/views/bs.view.show_block_slider.php* – шаблон блока баннеров.

Файловый архив

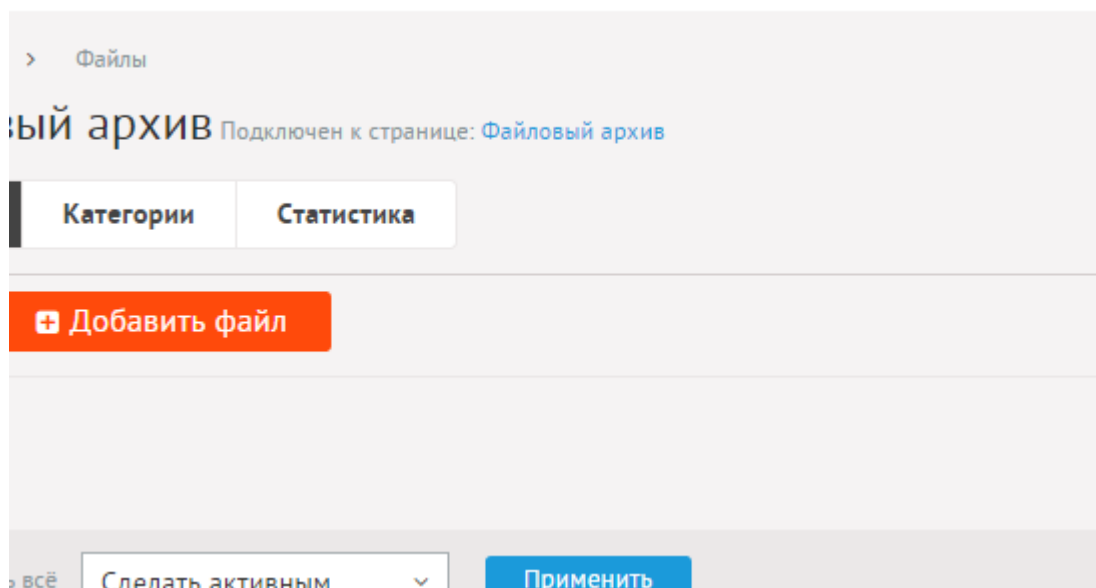
- [Файлы](#)
- [Категории](#)
- [Статистика](#)
- [Настройки](#)
- [Шаблонные теги](#)
- [База данных](#)
- [Файлы](#)

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

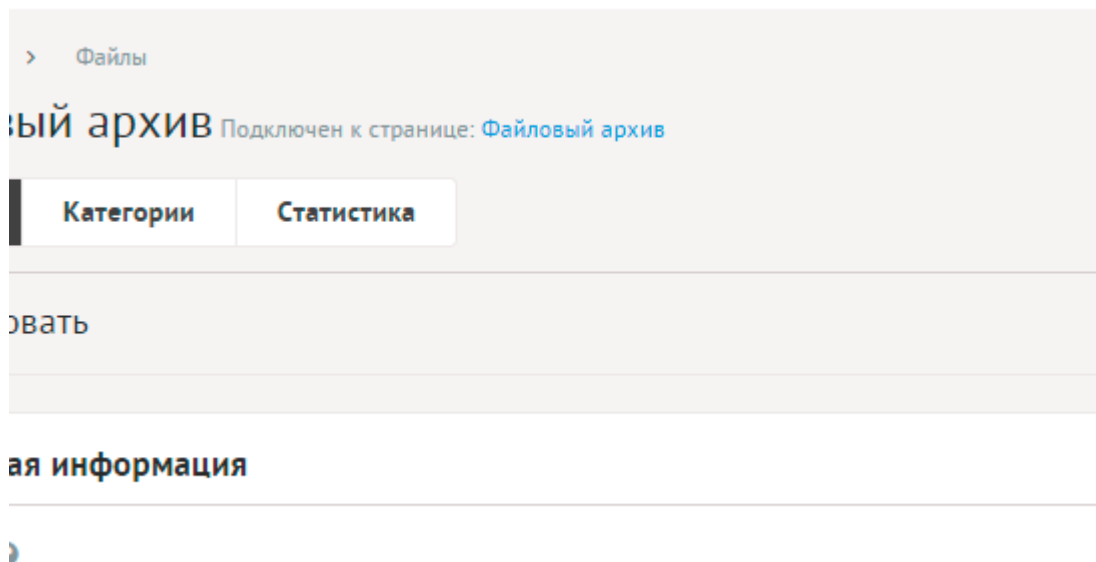
Модуль состоит из двух частей: категории файлов и файлы.

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



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).



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».
- **Категория** – Категория, к которой относится файл. Список категорий редактируется во вкладке выше. 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».
- **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»](#). Опция «HTML-код» позволяет отключить визуальный редактор для текущего поля. Значение этой настройки будет учитываться и при последующем редактировании. 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»](#). Опция «HTML-код» позволяет отключить визуальный редактор для текущего поля. Значение этой настройки будет учитываться и при последующем редактировании. Параметр имеет разные значения для разных языковых версий сайта.

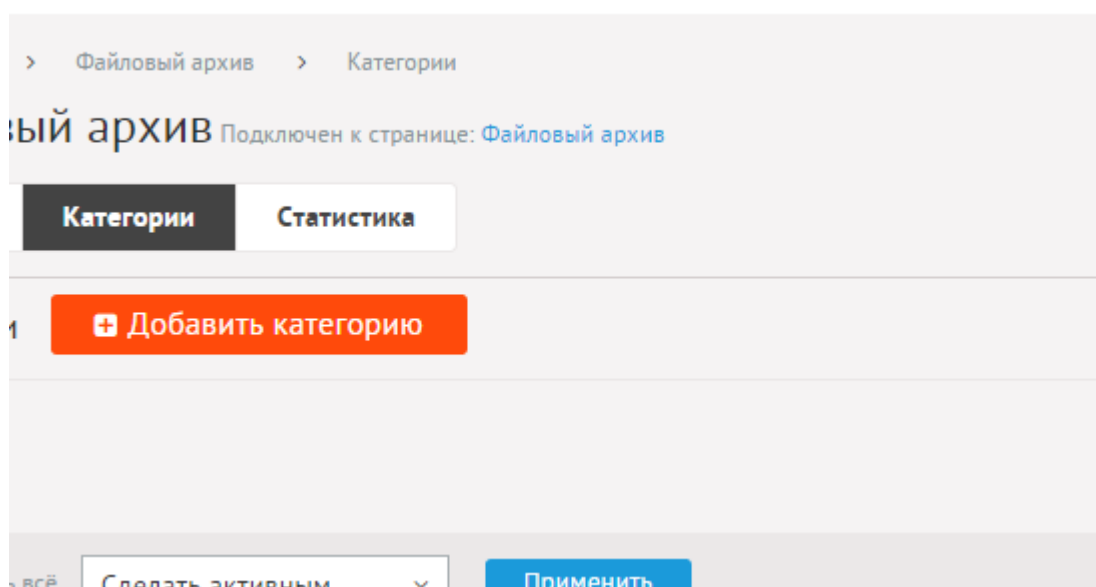
- **Динамические блоки.**
- **Теги** – Добавление тегов к файлу. Можно добавить либо новый тег, либо открыть и выбрать из уже существующих тегов. Параметр выводится, если в настройках модуля включен параметр «Подключить теги».
- **Похожие файлы** – Выбор и добавление к текущему файлу связей с другими 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.
- **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».

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*.
- **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.
- **Параметры SEO.**
- **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.

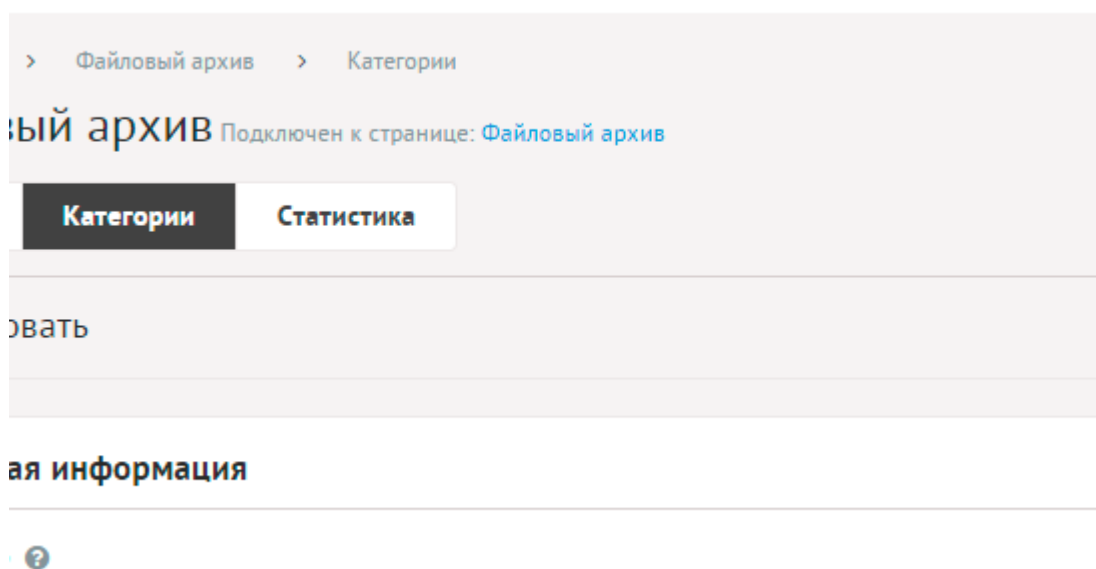
- **Канонический тег** – URL канонической страницы вида: <http://site.ru/pseudossylka/>, на которую переносится "ссылочный вес" данной страницы. Используется для страниц с похожим или дублирующимся контентом (SEO-специалисту). Параметр имеет разные значения для разных языковых версий сайта.
- **Псевдоссылка** – 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.
- **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).
- **Параметры показа.**
- **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»](#).
- **Оформление.**
- **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!).
- **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*.

Категории



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

При удалении категории удаляются все вложенные подкатегории и файлы.



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».

- **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»](#). Опция «HTML-код» позволяет отключить визуальный редактор для текущего поля. Значение этой настройки будет учитываться и при последующем редактировании. 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»](#). Опция «HTML-код» позволяет отключить визуальный редактор для текущего поля. Значение этой настройки будет учитываться и при последующем редактировании. Параметр имеет разные значения для разных языковых версий сайта.
- **Динамические блоки.**
- **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».
- **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*.

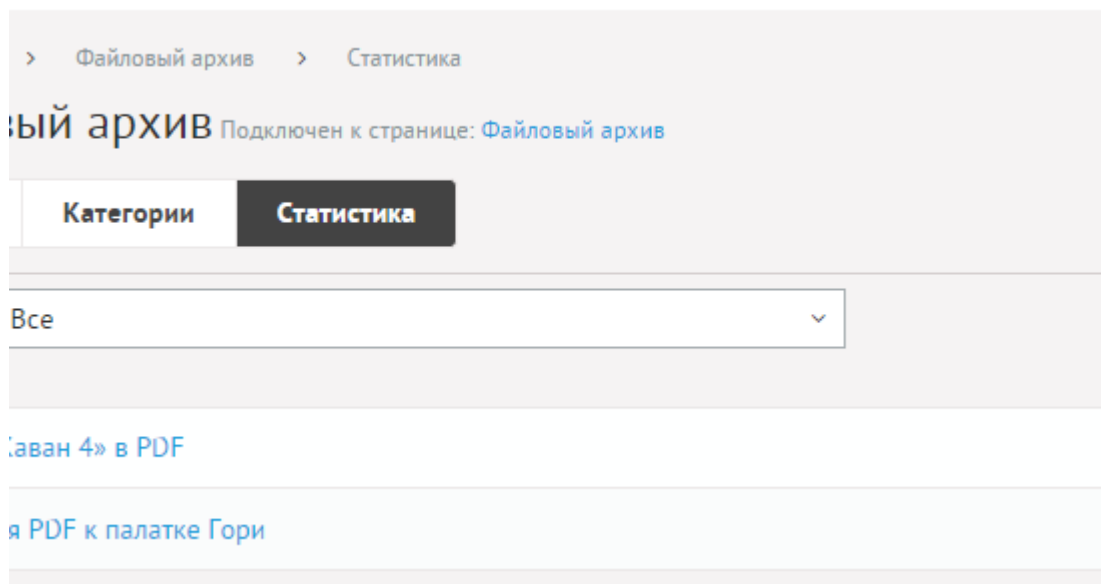
Дополнительные:

- **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*.
- **Параметры SEO.**
- **Заголовок окна в браузере, тег Title** – Если не заполнен, тег *Title* будет автоматически сформирован как «Название категории – 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, 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.

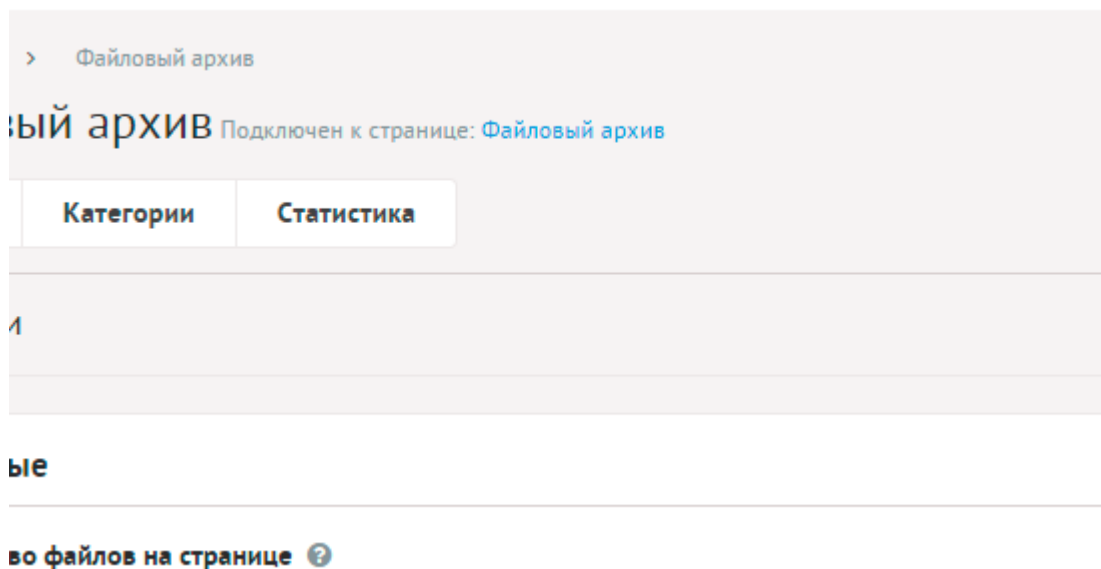
- **Канонический тег** – URL канонической страницы вида: <http://site.ru/pseudossylka/>, на которую переносится "ссылочный вес" данной страницы. Используется для страниц с похожим или дублирующимся контентом (SEO-специалисту). Параметр имеет разные значения для разных языковых версий сайта.
- **Псевдоссылка** – 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.
- **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).
- **Параметры показа.**
- **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).
- **Оформление.**
- **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 списка элементов** – Шаблон вывода контента модуля на странице элементов списка в категории (веб-мастеру и программисту, не меняйте этот параметр, если не уверены в результате!). Значение параметра важно для AJAX.
- **Шаблон страницы элемента** – 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!).

Статистика



Настройки



- **Основные.**
- **Количество файлов на странице** – Количество одновременно выводимых файлов в списке.
- **Включить «Показать ещё»** – На странице файлов появится кнопка «Показать ещё». Увеличивает количество одновременно выводимых файлов в списке.
- **Доступные типы файлов (через запятую).**

Пример:

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

- **Категории.**

- **Использовать категории** – Позволяет включить/отключить категории файлов.
- **Количество категорий на странице** – Количество одновременно выводимых категорий в списке на первой страницы модуля. Параметр появляется, если отмечена опция «Использовать категории».
- **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».
- **Изображения.**
- **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».
- **Подключения.**
- **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.
- **Автогенерация для SEO.**
- **Шаблон для автоматического генерирования Title** – Если шаблон задан и для файла не прописан заголовок *Title*, то заголовок автоматически генерируется по шаблону. В шаблон можно добавить:
 - %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 и для категории не прописан заголовок *Title*, то заголовок автоматически генерируется по шаблону. В шаблон можно добавить:
 - %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 и для файла не заполнено поле *Keywords*, то поле *Keywords* автоматически генерируется по шаблону. В шаблон можно добавить:

- %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 и для категории не заполнено поле *Keywords*, то поле *Keywords* автоматически генерируется по шаблону. В шаблон можно добавить:

- %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 и для файла не заполнено поле *Description*, то поле *Description* автоматически генерируется по шаблону. В шаблон можно добавить:

- %name – название,
- %category – название категории,
- %parent_category – название категории верхнего уровня,
- %anons – краткое описание (SEO-специалисту).

Параметр имеет разные значения для разных языковых версий сайта.

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

- %name – название категории,
- %parent – название категории верхнего уровня,
- %anons – краткое описание (SEO-специалисту).

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

- **Оформление.**

- **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 элементов в списке** – По умолчанию *modules/files/views/files.view.rows.php*. Параметр для разработчиков! Не устанавливайте, если не уверены в результате. Значение параметра важно для AJAX.

- **Шаблон для первой страницы модуля (если подключены категории)** – По умолчанию *modules/files/views/files.view.ftst_page.php*. Параметр для разработчиков! Не устанавливайте, если не уверены в результате.

- **Шаблон для элементов в списке первой страницы модуля (если подключены категории)** – По умолчанию *modules/files/views/files.view.ftst_page.php*. Параметр для разработчиков! Не устанавливайте, если не уверены в результате. Значение параметра важно для AJAX.

- **Шаблон для страницы элемента** – По умолчанию, *modules/files/views/files.view.id.php*. Параметр для разработчиков! Не устанавливайте, если не уверены в результате.
- **Дополнительно.**
- **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. Можно указать отрицательное значение, тогда будут исключены файлы из указанного раздела. По умолчанию выбираются все страницы;
- **cat_id** – категории файлов, если в настройках модуля отмечено «Использовать категории». Идентификаторы категорий перечисляются через запятую. Можно указать отрицательное значение, тогда будут исключены файлы из указанной категории. Можно указать значение **current**, тогда будут показаны файлы из текущей (открытой) категории или из всех категорий, если ни одна категория не открыта. По умолчанию категория не учитывается, выводятся все файлы;
- **sort** – сортировка файлов: по умолчанию как на странице модуля, **rand** – в случайном порядке, **date** – по дате;
- **images** – количество изображений, прикрепленных к файлу;
- **images_variation** – тег размера изображений, задается в настройках модуля;
- **only_module** – выводить блок только на странице, к которой прикреплен модуль «Файловый архив»: **true** – выводить блок только на странице модуля, по умолчанию блок будет выводиться на всех страницах;
- **tag** – тег, прикрепленный к файлам;
- **defer** – маркер отложенной загрузки шаблонного тега: **event** – загрузка контента только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** – загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка

контента только по желанию пользователя;

— **defer_title** – текстовая строка, выводимая на месте появления загружаемого контента с помощью отложенной загрузки шаблонного тега;

— **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

В шаблоне тега можно получить значение любого атрибута через переменную `$result["attributes"]["название атрибута"]`.

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;

— **defer** – маркер отложенной загрузки шаблонного тега: **event** – загрузка контента только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** – загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка контента только по желанию пользователя;

— **defer_title** – текстовая строка, выводимая на месте появления загружаемого контента с помощью отложенной загрузки шаблонного тега;

— **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

В шаблоне тега можно получить значение любого атрибута через переменную

`$result["attributes"]["название атрибута"]`.

База данных

[`{files}`](#) – Файлы в файловом архиве

[`{files_category}`](#) – Категории файлового архива

[`{files_category_parents}`](#) – Родительские связи категорий файлового архива

[`{files_category_rel}`](#) – Связи файлов и категорий в файловом архиве

[`{files_counter}`](#) – Счетчик просмотров файлов в файловом архиве

[`{files_links}`](#) – Внешние ссылки на файлы для файлового архива

[`{files_rel}`](#) – Связи похожих файлов

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/admin/js/files.admin.js` – редактирование файлов, JS-сценарий;
7. `modules/files/files.php` – контроллер;
8. `modules/files/files.install.php` – установка модуля;
9. `modules/files/files.model.php` – модель;
10. `modules/files/files.search.php` – настройки для поисковой индексации для модуля «Поиск»;
11. `modules/files/files.sitemap.php` – карта ссылок для модуля «Карта сайта»;
12. `modules/files/views/files.view.first_page.php` – шаблон первой страницы модуля, если в настройках модуля подключен параметр «Использовать категории»;
13. `modules/files/views/files.view.id.php` – шаблон страницы файла;
14. `modules/files/views/files.view.list.php` – шаблон вывода списка файлов;
15. `modules/files/views/files.view.rows.php` – шаблон вывода списка файлов;

16. *modules/files/views/files.view.rows_block.php* – шаблон блока файлов;
17. *modules/files/views/files.view.rows_block_rel.php* – шаблон блока похожих файлов;
18. *modules/files/views/files.view.rows_first_page.php* – шаблон вывода списка файлов;
19. *modules/files/views/files.view.show_block.php* – шаблон блока файлов;
20. *modules/files/views/files.view.show_block_rel.php* – шаблон блока похожих файлов.

Объявления

- [Объявления](#)
- [Категории](#)
- [Характеристики](#)
- [Статистика](#)
- [Настройки](#)
- [Шаблонные теги](#)
- [База данных](#)
- [Файлы](#)

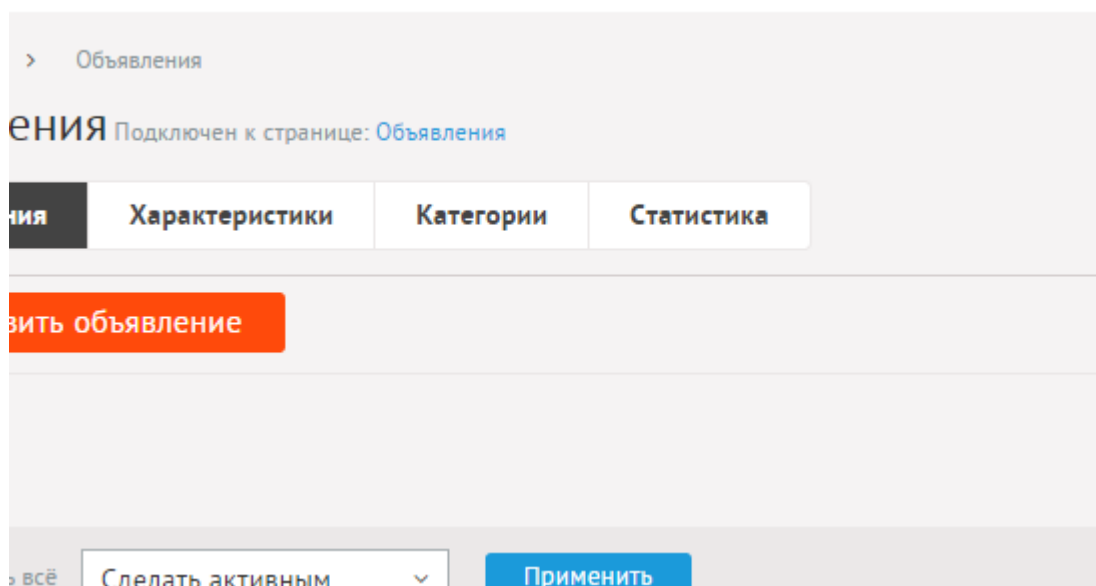
Модуль «Объявления» отображает текстово-графические материалы с сортировкой по датам.

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

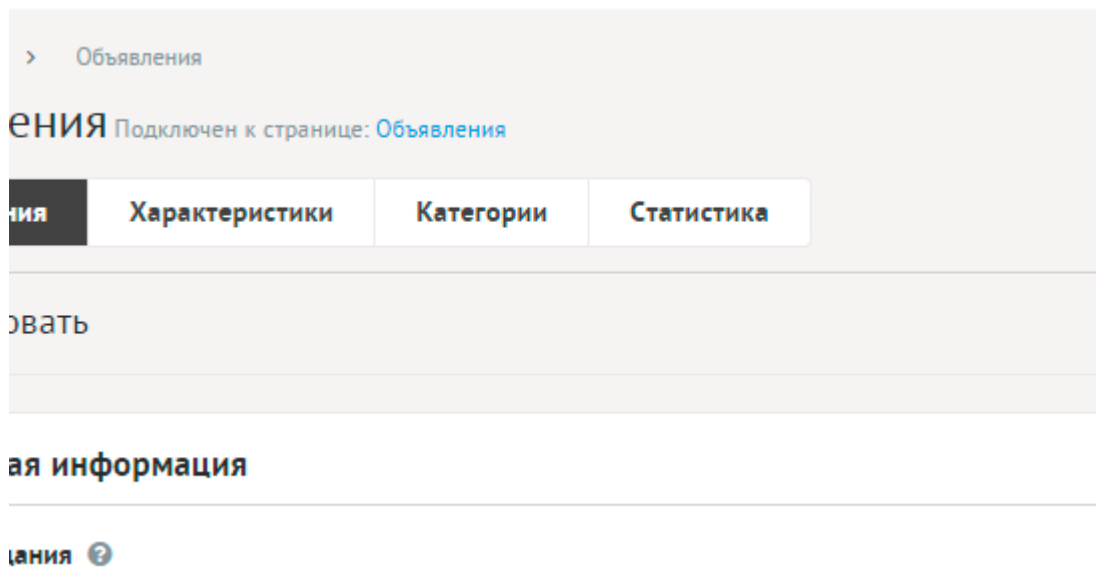
Модуль «Объявления» можно прикреплять к различным страницам сайта. Если модуль прикреплен к нескольким страницам, в административной части модуля появляется специальный фильтр «Раздел сайта» с возможностью фильтрации вывода (списка объявлений, категорий и настроек) по принадлежности к странице.

Модуль состоит из трёх частей: категории объявлений, объявления и характеристики объявлений.

Объявления



В списке объявления сортируются по дате.



Объявление имеет следующие характеристики.

Основные:

- **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.
- **Автор объявления** – Пользователь, создавший объявление в форме на сайте.
- **Заголовок** – Используется в ссылках на объявление, заголовках. Параметр имеет разные значения для разных языковых версий сайта.
- **Опубликовать на сайте** – Если не отмечена, объявление не будет отображаться на сайте. Параметр имеет разные значения для разных языковых версий сайта.
- **Категория** – Категория, к которой относится объявление. Список категорий редактируется во вкладке выше. 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».
- **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.**
- **Характеристики** – Группа полей, определенных в части «Характеристики». Параметр имеет разные значения для разных языковых версий сайта.
- **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»](#). Опция «HTML-код» позволяет отключить визуальный редактор для текущего поля. Значение этой настройки будет

учитываться и при последующем редактировании. 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»](#). Опция «HTML-код» позволяет отключить визуальный редактор для текущего поля. Значение этой настройки будет учитываться и при последующем редактировании. Параметр имеет разные значения для разных языковых версий сайта.
- **Динамические блоки.**
- **Точка на карте** – Возможность установить или отредактировать точку на геокарте. Параметр выводится, если в настройках модуля включен параметр «Подключить геокарту».
- **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.
- **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».
- **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».

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*.
- **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.
- **Параметры SEO.**
- **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.
- **Канонический `тег`** – URL канонической страницы вида: `http://site.ru/pseudossylka/`, на которую переносится "ссылочный вес" данной страницы. Используется для страниц с похожим или дублирующимся контентом (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.
- **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).
- **Параметры показа.**
- **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, объявление 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»](#).
- **Оформление.**
- **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!).
- **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*.

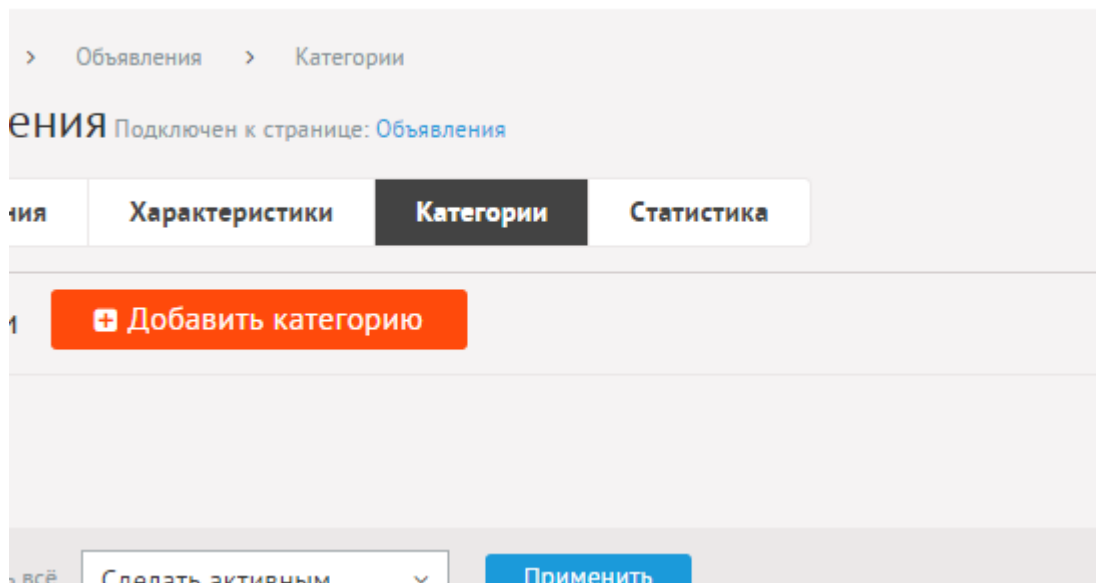
В модуль «Объявления» встроена автоматическая публикация объявлений в формате RSS.

Адрес RSS-ленты: *http://site.ru/ab/rss/*.

У англоязычной версии сайта адрес RSS-ленты новостей будет иметь вид:

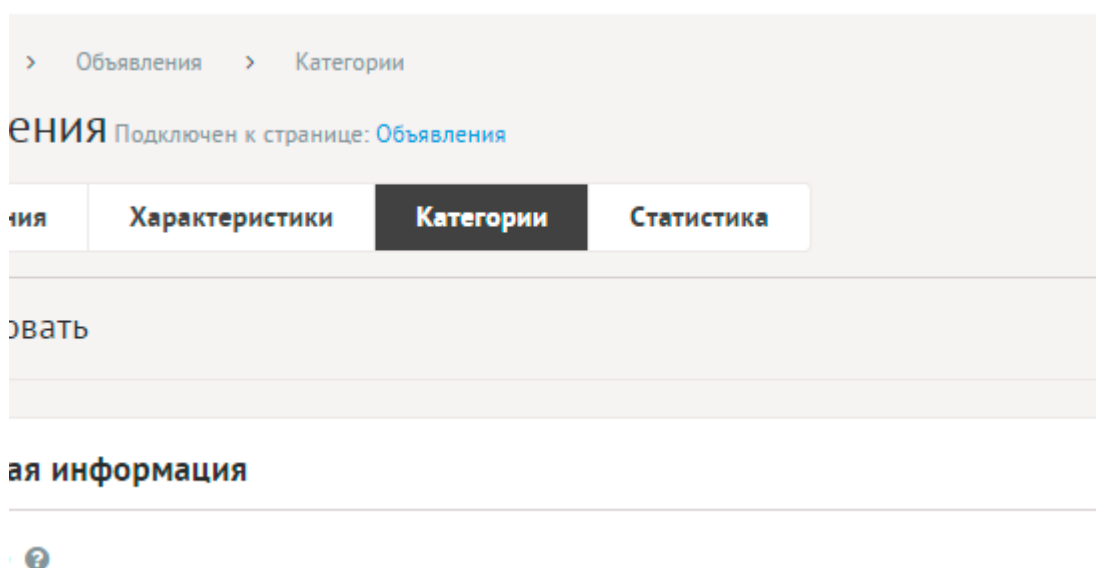
http://site.ru/eng/ab/rss/

Категории



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

При удалении категории удаляются все вложенные подкатегории и объявления.



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, прикрепленные к категории** – Список характеристик, применимых к объявлениям текущей категории.
- **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»](#). Опция «HTML-код» позволяет отключить визуальный редактор для текущего поля. Значение этой настройки будет учитываться и при последующем редактировании. 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»](#). Опция «HTML-код» позволяет отключить визуальный редактор для текущего поля. Значение этой настройки будет учитываться и при последующем редактировании. Параметр имеет разные значения для разных языковых версий сайта.
- **Динамические блоки.**
- **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».
- **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:

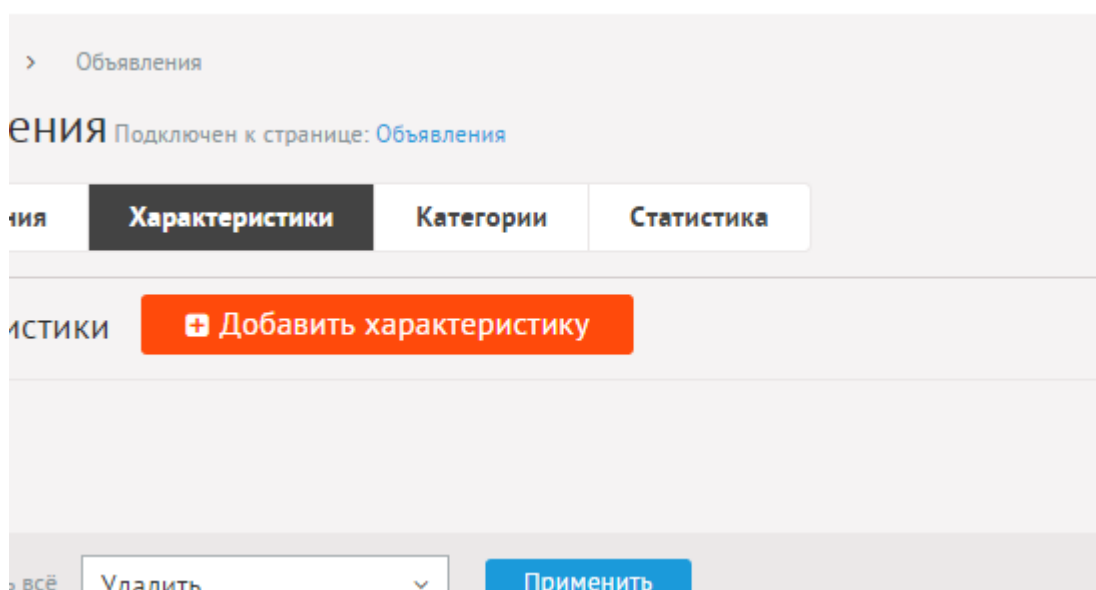
- **Номер** – Номер элемента в БД (веб-мастеру и программисту).
- **Редактор** – Изменяется после первого сохранения. Показывает, кто из администраторов сайта первый правил текущую страницу.
- **Время последнего изменения** – Изменяется после сохранения элемента. Отдается в

заголовке *Last Modify*.

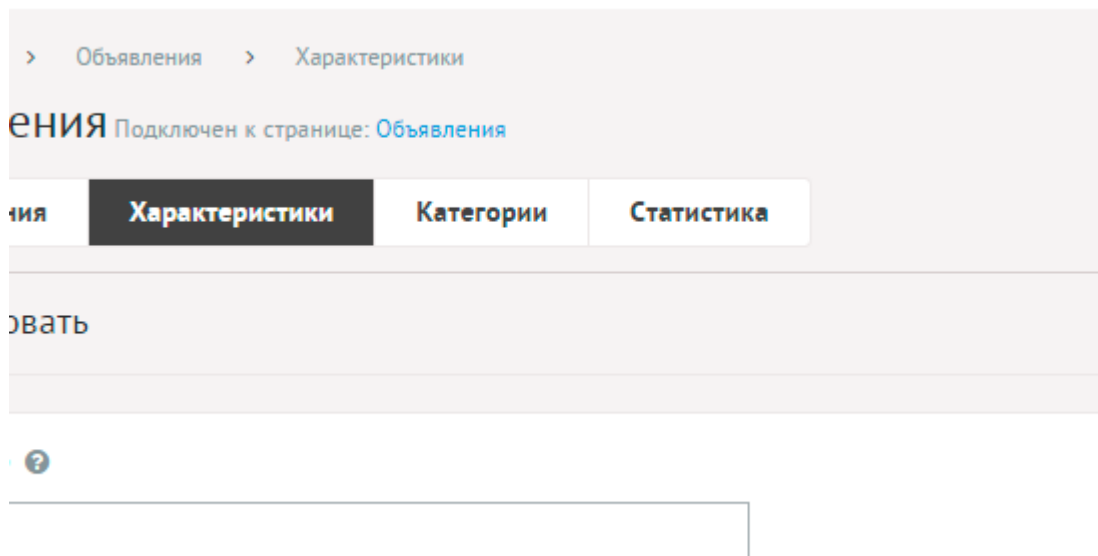
- **Раздел сайта** – Перенос категории на другую страницу сайта, к которой прикреплен модуль (администратору сайта).
- **Параметры SEO.**
- **Заголовок окна в браузере, тег Title** – Если не заполнен, тег *Title* будет автоматически сформирован как «Название категории – Название страницы – Название сайта», 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, 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.
- **Канонический тег** – URL канонической страницы вида: <http://site.ru/pseudossylka/>, на которую переносится "ссылочный вес" данной страницы. Используется для страниц с похожим или дублирующимся контентом (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.
- **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).
- **Параметры показа.**
- **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).
- **Оформление.**
- **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 списка элементов** – Шаблон вывода контента модуля на странице элементов списка в категории (веб-мастеру и программисту, не меняйте этот параметр, если не уверены в результате!). Значение параметра важно для AJAX.
- **Шаблон страницы элемента** – 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).

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.



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,
 - поле с визуальным редактором,
 - электронный ящик,
 - телефон,
 - 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».
- **Раздел сайта** – Раздел сайта, к которому применяется характеристика.
- **Категория** – Категории объявлений, для которых действует данный параметр. Если не указана ни одна категория, то характеристика считается общей.
- **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.
- **Показывать на странице объявления** – Если отмечено, данная характеристика будет отображаться на странице объявления.
- **Отображать параметры в блоке для сортировки объявлений** – Позволяет выводить характеристику в виде ссылки для сортировки объявлений по значению

характеристики.

- **Описание характеристики** – Параметр имеет разные значения для разных языковых версий сайта.
- **Сортировка: установить перед** – Редактирование порядка следования характеристики в списке

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.

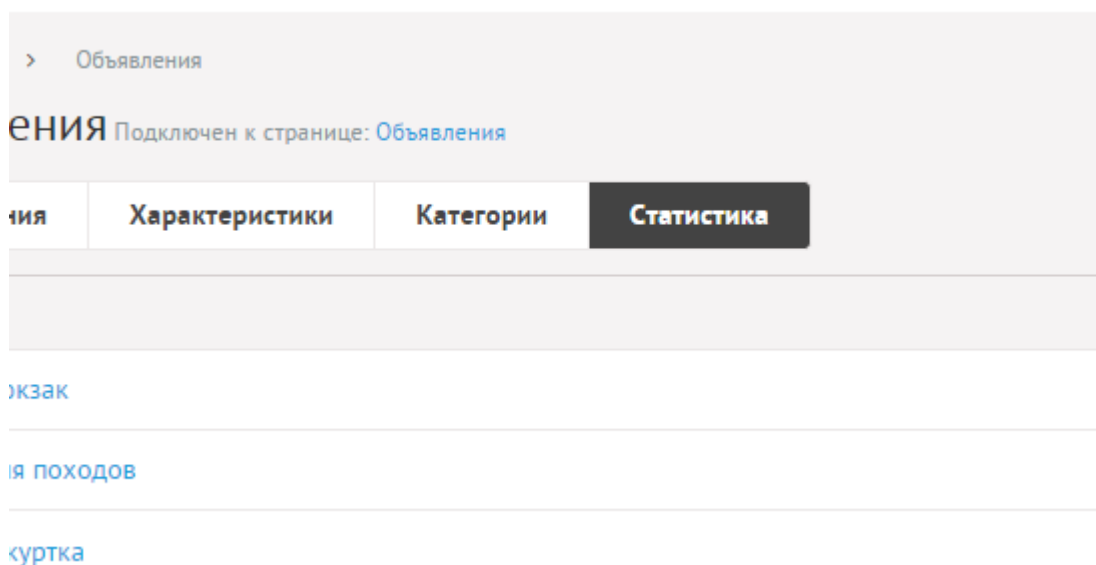
Пример:

*Посредник: да
Посредник: нет.*

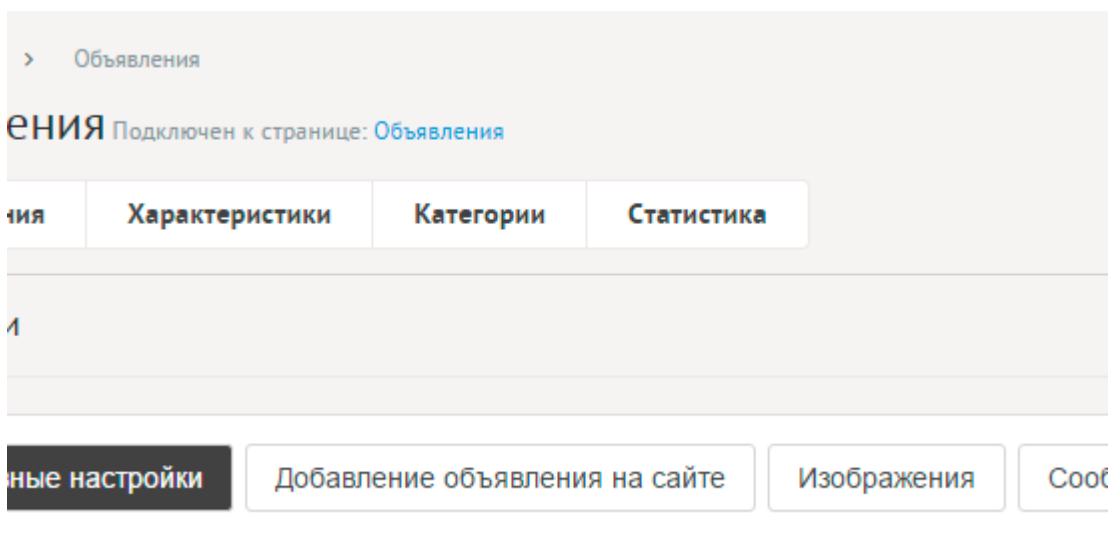
Если выбран тип «выпадающий список» или «список с выбором нескольких значений», то появятся дополнительные поля со значениями.

Если выбран тип «поле с визуальным редактором», то при редактировании поля будет отображаться визуальный редактор, что позволяет редактировать стиль текста.

Статистика



Настройки



Основные настройки

- **Основные.**
- **Количество объявлений на странице** – Количество одновременно выводимых объявлений в списке.
- **Количество категорий на странице** – Количество одновременно выводимых категорий в списке на первой страницы модуля. Параметр появляется, если отмечена опция «Использовать категории».
- **Включить «Показать ещё»** – На странице объявлений появится кнопка «Показать ещё». Увеличивает количество одновременно выводимых объявлений в списке.
- **Формат даты** – Позволяет настроить отображение даты в модуле. Возможные значения:
 - 01.05.2016,
 - 01.05.2016 14:45,
 - 1 мая 2016 г.,
 - 1 мая,
 - 1 мая 2016, понедельник,
 - вчера 15:30,
 - не отображать.
- **В блоке похожих объявлений связь двусторонняя** – Позволяет установленную в объявлении связь с другим объявлением использовать в двух направлениях.
- **Категории.**
- **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».
- **Подключения.**
- **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 геокарту** – Подключение модуля «Геокарта». Параметр не будет включен, если модуль «Геокарта» не установлен. Подробности см. в разделе [модуль «Геокарта»](#).
- **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»](#).
- **Автогенерация для SEO.**
- **Шаблон для автоматического генерирования Title** – Если шаблон задан и для объявления не прописан заголовок *Title*, то заголовок автоматически генерируется по шаблону. В шаблон можно добавить:
 - %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 и для категории не прописан заголовок *Title*, то заголовок автоматически генерируется по шаблону. В шаблон можно добавить:
 - %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 и для объявления не заполнено поле *Keywords*, то поле *Keywords* автоматически генерируется по шаблону. В шаблон можно добавить:

- %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 и для категории не заполнено поле *Keywords*, то поле *Keywords* автоматически генерируется по шаблону. В шаблон можно добавить:

- %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 и для объявления не заполнено поле *Description*, то поле *Description* автоматически генерируется по шаблону. В шаблон можно добавить:

- %name – название,
- %category – название категории,
- %parent_category – название категории верхнего уровня,
- %anons – краткое описание (SEO-специалисту).

Параметр имеет разные значения для разных языковых версий сайта.

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

- %name – название категории,
- %parent – название категории верхнего уровня,
- %anons – краткое описание (SEO-специалисту).

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

- **Оформление.**
- **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 элементов в списке** – По умолчанию *modules/ab/views/ab.view.rows.php*. Параметр для разработчиков! Не устанавливайте, если не уверены в результате. Значение параметра важно для AJAX.
- **Шаблон для первой страницы модуля (если подключены категории)** – По умолчанию *modules/ab/views/ab.view.ftst_page.php*. Параметр для разработчиков! Не устанавливайте, если не уверены в результате.
- **Шаблон для элементов в списке первой страницы модуля (если подключены категории)** – По умолчанию *modules/ab/views/ab.view.ftst_page.php*. Параметр для разработчиков! Не устанавливайте, если не уверены в результате. Значение

параметра важно для AJAX.

- **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 элементов списка с одинаковой характеристикой** – Параметр для разработчиков! Не устанавливайте, если не уверены в результате. Значение параметра важно для AJAX.
- **Шаблон для поиска элементов** – Параметр для разработчиков! Не устанавливайте, если не уверены в результате.
- **Шаблон элементов в списке для поиска элементов** – Параметр для разработчиков! Не устанавливайте, если не уверены в результате. Значение параметра важно для AJAX.
- **Шаблон для объявлений пользователя** – Параметр для разработчиков! Не устанавливайте, если не уверены в результате.
- **Шаблон для объявлений пользователя** – Параметр для разработчиков! Не устанавливайте, если не уверены в результате. Значение параметра важно для AJAX.
- **Дополнительно.**
- **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.
- **Модерация объявлений для** – Объявление добавляется из формы на сайте не активным и активируется в административной части сайта.
- **Использовать в форме объявления обязательные поля** – Позволяет выводить/скрывать 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 перекодировать размер уже загруженных изображений. Кнопка необходима, если изменены настройки размеров изображений. Параметр выводится, если отмечена опция «Использовать изображения».

Сообщения и уведомления

- **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>Объявление успешно добавлено.</b></div>
```

- **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».

Пример:

Новое объявление на сайте %title (%url)

- **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».

Пример:

Здравствуйте, администратор сайта %title (%url)!
В рубрике Объявления появилось новое объявление:
%message

- **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. Можно указать отрицательное значение, тогда будут исключены объявления из указанного раздела. По умолчанию выбираются все

страницы;

- **cat_id** – категории объявлений, если в настройках модуля отмечено «Использовать категории». Идентификаторы категорий перечисляются через запятую. Можно указать отрицательное значение, тогда будут исключены объявления из указанной категории. Можно указать значение **current**, тогда будут показаны объявления из текущей (открытой) категории или из всех категорий, если ни одна категория не открыта. По умолчанию категория не учитывается, выводятся все объявления;
- **sort** – сортировка объявлений: **date** – по дате (по умолчанию), **rand** – в случайном порядке;
- **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 *param*="3=5&3=6&10>50&16=1" расшифровывается как объявления о квартире и доме (5 и 6 номер), площадью более 50 кв.м, в новостройке.

```
<insert name="show_block" module="ab" param="3=5&3=6&10>50&16=1">
```

Номер (или идентификатор) характеристики можно посмотреть, если подвести курсор к названию характеристики в списке характеристик в административной части. Появится всплывающая подсказка «Редактировать (номер характеристики)».

Номер (или идентификатор) значения характеристики можно посмотреть, если при редактировании характеристики подвести курсора на нужное значение. Появится всплывающая подсказка «ID: номер».

- **only_module** – выводить блок только на странице, к которой прикреплен модуль «Объявления»: **true** – выводить блок только на странице модуля, по умолчанию блок будет выводиться на всех страницах;
- **tag** – тег, прикрепленный к объявлениям;
- **defer** – маркер отложенной загрузки шаблонного тега: **event** – загрузка контента только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** –

загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка контента только по желанию пользователя;

— **defer_title** – текстовая строка, выводимая на месте появления загружаемого контента с помощью отложенной загрузки шаблонного тега;

— **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

В шаблоне тега можно получить значение любого атрибута через переменную `$result["attributes"]["название атрибута"]`.

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 объявлению;
- **images_variation** – тег размера изображений, задается в настройках модуля;
- **defer** – маркер отложенной загрузки шаблонного тега: **event** – загрузка контента только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** – загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка контента только по желанию пользователя;

- **defer_title** – текстовая строка, выводимая на месте появления загружаемого контента с помощью отложенной загрузки шаблонного тега;
- **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

В шаблоне тега можно получить значение любого атрибута через переменную `$result["attributes"]["название атрибута"]`.

show_form – выводит форму добавления сообщения. Для правильной работы тега должна существовать страница, к которой прикреплен модуль Объявления.

Атрибуты:

- **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»;
- **only_module** – выводить форму только на странице, к которой прикреплен модуль «Объявления»: **true** – выводить форму только на странице модуля, по умолчанию форма будет выводиться на всех страницах;
- **defer** – маркер отложенной загрузки шаблонного тега: **event** – загрузка контента только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** – загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка контента только по желанию пользователя;
- **defer_title** – текстовая строка, выводимая на месте появления загружаемого контента с помощью отложенной загрузки шаблонного тега;
- **template** – шаблон тега (файл *modules/ab/views/ab.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

В шаблоне тега можно получить значение любого атрибута через переменную `$result["attributes"]["название атрибута"]`.

show_search – выводит форму поиска объявлений. Если для категорий прикреплены дополнительные характеристики, то поиск по ним производится только на странице категории.

Атрибуты:

- **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. Можно указать отрицательное значение, тогда указанные страницы будут исключены из списка;
- **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 отрицательное значение, тогда указанные категории будут исключены из списка. Можно указать значение **all**, тогда поиск будет осуществляться по всем категориям объявлений и в форме будут участвовать только общие характеристики. Атрибут не обязателен;
- **ajax** – подгружать результаты поиска без перезагрузки страницы. Результаты подгружаются только если открыта страница со списком объявлений, иначе поиск работает обычным образом: **true** – результаты поиска подгружаются, по умолчанию будет перезагружена вся страница;
- **only_module** – выводить блок только на странице, к которой прикреплен модуль «Объявления»: **true** – выводить блок только на странице модуля, по умолчанию блок будет выводиться на всех страницах;
- **defer** – маркер отложенной загрузки шаблонного тега: **event** – загрузка контента только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** – загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка контента только по желанию пользователя;

- **defer_title** – текстовая строка, выводимая на месте появления загружаемого контента с помощью отложенной загрузки шаблонного тега;
- **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

В шаблоне тега можно получить значение любого атрибута через переменную `$result["attributes"]["название атрибута"]`.

База данных

[{ab}](#) – Объявления

[{ab_category}](#) – Категории объявлений

[{ab_category_parents}](#) – Родительские связи категорий объявлений

[{ab_category_rel}](#) – Связи объявлений и категорий

[{ab_counter}](#) – Счетчик просмотров объявлений

[{ab_param}](#) – Дополнительные характеристики объявлений

[{ab_param_category_rel}](#) – Связи дополнительных характеристик объявлений и категорий

[{ab_param_element}](#) – Значений дополнительных характеристик объявлений

[{ab_param_select}](#) – Варианты значений дополнительных характеристик объявлений типа список

[{ab_rel}](#) – Связи похожих объявлений

Файлы

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

1. *modules/ab/ab.php* – контроллер;
2. *modules/ab/ab.action.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.action.php* – обработка POST-запросов в административной части модуля;
10. *modules/ab/admin/ab.admin.category.php* – редактирование категорий объявлений;
11. *modules/ab/admin/ab.admin.config.php* – настройки модуля;
12. *modules/ab/admin/ab.admin.count.php* – количество новых и неактивных объявлений для меню административной панели;
13. *modules/ab/admin/ab.admin.counter.php* – статистика просмотров;
14. *modules/ab/admin/ab.admin.menu.php* – карта ссылок для модуля «Меню на сайте»;
15. *modules/ab/admin/ab.admin.param.php* – редактирование дополнительных характеристик объявлений;
16. *modules/ab/admin/js/ab.admin.param.js* – редактирование характеристик объявлений, JS-сценарий;
17. *modules/ab/img/** – изображения;
18. *modules/ab/js/ab.edit.js* – JS-сценарий формы редактирования объявления;
19. *modules/ab/js/ab.form.js* – JS-сценарий формы добавления объявления;
20. *modules/ab/js/ab.js* – JS-сценарий модуля;
21. *modules/ab/js/ab.show_search.js* – JS-сценарий формы поиска по объявлениям;
22. *modules/ab/views/ab.view.attachments.php* – шаблон прикрепленных к объявлению файлов в форме редактирования объявления;
23. *modules/ab/views/ab.view.edit.php* – шаблон формы редактирования объявления;
24. *modules/ab/views/ab.view.first_page.php* – шаблон вывода первой страницы модуля, если в настройках модуля подключен параметр «Использовать категории»;
25. *modules/ab/views/ab.view.form.php* – шаблон формы добавления объявления;
26. *modules/ab/views/ab.view.id.php* – шаблон страница объявления;
27. *modules/ab/views/ab.view.images.php* – шаблон прикрепленных к объявлению изображения, выводимых в форме редактирования объявления;
28. *modules/ab/views/ab.view.list.php* – шаблон список объявлений;

- 29. *modules/ab/views/ab.view.param.php* – шаблон дополнительных характеристик объявлений;
- 30. *modules/ab/views/ab.view.rows.php* – шаблон элементов в списке объявлений;
- 31. *modules/ab/views/ab.view.rows_block.php* – шаблон блока объявлений;
- 32. *modules/ab/views/ab.view.rows_block_rel.php* – шаблон блока похожих объявлений;
- 33. *modules/ab/views/ab.view.show_block.php* – шаблон блока объявлений;
- 34. *modules/ab/views/ab.view.show_block_rel.php* – шаблон блока похожих объявлений;
- 35. *modules/ab/views/ab.view.show_search.php* – шаблон форма поиска по объявлениям;
- 36. *modules/ab/views/ab.view.sort_block.php* – шаблон блока «Сортировать» с ссылками на направление сортировки.

Теги

- [Подключение](#)
- [Как добавлять теги к модулю](#)
- [Административная часть](#)
- [Настройки](#)
- [Шаблонные теги](#)
- [База данных](#)
- [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|boolean false get ([integer $element_id = 0], [string $module_name = ''], [string $element_type = 'element'], [integer $site_id = 0])` – Выводит подключенные к элементу теги.

- `integer $element_id`: номер элемента, к которому прикреплены теги, по умолчанию текущий элемент модуля
- `string $module_name`: название модуля, по умолчанию модуль, прикрепленный к текущей странице сайта
- `string $element_type`: тип данных
- `integer $site_id`: страница сайта, к которой прикреплен элемент, по умолчанию текущая страница сайта

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;
```

`void prepare ([integer $element_id = 0], [string $module_name = ''], [string $element_type = 'element'])` – Запоминает данные элемента, которому нужно будет вывести теги.

- `integer $element_id`: номер элемента, к которому прикреплены теги, по умолчанию текущий элемент модуля
- `string $module_name`: название модуля, по умолчанию модуль, прикрепленный к

текущей странице сайта

- string \$element_type: тип данных

Пример:

// в этом примере будет выполнено три SQL-запроса к базе данных для получения тегов ко всем заданным статьям

```
$ids = array(3, 5, 7);
foreach($ids as $id)
{
    $tags[$id] = $this->diafan->_tags->get($id, 'clauses');
}
```

Пример:

// в этом примере будет выполнен один SQL-запрос к базе данных для получения тегов ко всем заданным статьям

```
$ids = array(3, 5, 7);
foreach($ids as $id)
{
    $this->diafan->_tags->prepare($id, 'clauses');
}
foreach($ids as $id)
{
    $tags[$id] = $this->diafan->_tags->get($id, 'clauses');
}
```

void **delete** (integer|array \$element_ids, string \$module_name, [string \$element_type = 'element']) – Удаляет теги для одного или нескольких элементов.

- integer|array \$element_ids: номер одного или нескольких элементов
- string \$module_name: название модуля
- string \$element_type: тип данных

Пример:

// удалит теги, прикрепленные к статье ID=3

```
$this->diafan->_tags->delete(3, 'clauses');
```

// удалит теги, прикрепленные к статьям ID=3,4,5

```
$this->diafan->_tags->delete(array(3, 4, 5) 'clauses');
```

void **delete_module** (string \$module_name) – Удаляет все теги модуля.

- string \$module_name: название модуля

Пример:

// удалит теги, прикрепленные ко всем статьям

```
$this->diafan->_tags->delete_module('clauses');
```

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 об элементах модуля оформляются так же, как в списке элементов этого модуля. Для этого данные обрабатываются функцией **elements()** из модели модуля *modules/модуль/модуль.model.php* и выводятся шаблоном *modules/модуль/views/модуль.view.list.php*.

Можно задать свою обработку данных для списка элементов, прикрепленных к тегу. Для этого нужно определить функцию **tags()** в файле *modules/модуль/модуль.model.php*.

Пример:

Так обработка данных о прикрепленных к тегу товарах идет через функцию **tags()** в файле *modules/shop/shop.model.php*.

А обработка данных о прикрепленных к тегу новостях идет через функцию **elements()** в файле *modules/news/news.model.php*, так как функцию **tags()** в этом файле не описана.

Шаблон также можно применить отличный от шаблона по умолчанию. Для этого достаточно создать файл *modules/модуль/views/модуль.view.list_tags.php*.

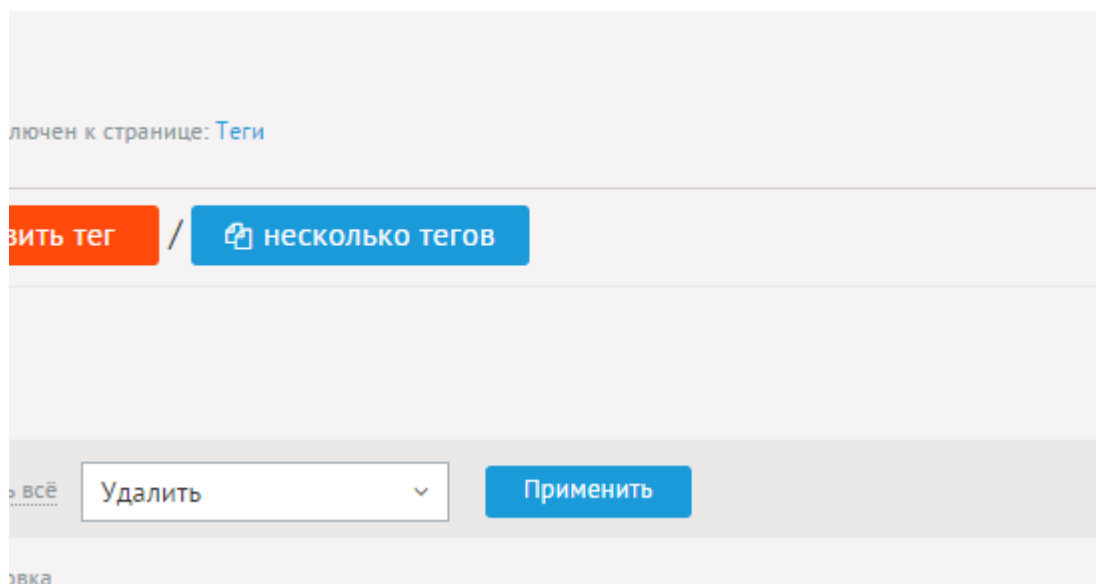
Пример:

Товары оформляются на странице тега шаблоном *modules/shop/views/shop.view.list_tags.php*.

А новости оформляются также, как в списке новостей шаблоном *modules/news/views/news.view.list.php*.

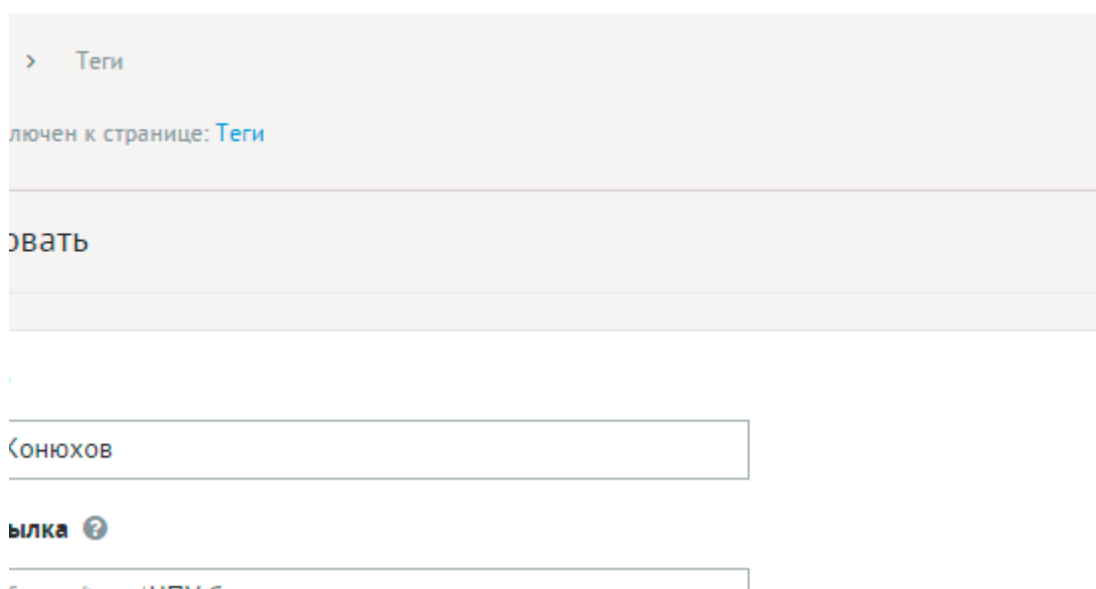
Описанные выше особенности относятся только к элементам модулей (новости, товары, статьи и пр.). Страницы сайта выводятся стандартным шаблоном *modules/tags/views/tags.view.list.php*.

Административная часть



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.

При удалении какого-либо элемента удаляются все теги, прикрепленные к нему.



Теги имеют следующие характеристики.

Основные:

- **Название** – 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.
- **Изображения** – Назначить тегу изображение.

- **Динамические блоки.**
- **Описание** – Параметр имеет разные значения для разных языковых версий сайта.

Дополнительные:

- **Время последнего изменения** – Изменяется после сохранения элемента. Отдается в заголовке *Last Modify*.
- **Параметры SEO.**
- **Заголовок окна в браузере, тег Title** – Если не заполнен, тег *Title* будет автоматически сформирован как «Названия тега – Название страницы – Название сайта» (SEO-специалисту). Параметр имеет разные значения для разных языковых версий сайта.
- **Ключевые слова, тег Keywords** – Параметр имеет разные значения для разных языковых версий сайта.
- **Описание, тег Description** – Параметр имеет разные значения для разных языковых версий сайта.
- **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).
- **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

> Теги

ключен к странице: Теги

1

во элементов на странице ?

пользовать изображения ?

- **Количество элементов на странице** – Количество одновременно выводимых

новостей, товаров, фотографий и др., помеченных тегом в списке.

- **Включить «Показать ещё»** – На странице элементов появится кнопка «Показать ещё». Увеличивает количество одновременно выводимых элементов 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».
- **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».

Шаблонные теги

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

show_block – выводит список тегов.

Атрибуты:

— **defer** – маркер отложенной загрузки шаблонного тега: **event** – загрузка контента только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** – загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка контента только по желанию пользователя;

— **defer_title** – текстовая строка, выводимая на месте появления загружаемого контента с помощью отложенной загрузки шаблонного тега;

— **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

В шаблоне тега можно получить значение любого атрибута через переменную `$result["attributes"]["название атрибута"]`.

База данных

[`{tags}`](#) – Связи тегов и элементов модулей

[`{tags_name}`](#) – Теги

Файлы

Модуль находится в папке *modules/tags*.

1. *modules/tags/admin/js/tags.admin.inc.js* – редактирование тегов, JS-сценарий;
2. *modules/tags/admin/js/tags.admin.js* – редактирование тегов, JS-сценарий;
3. *modules/tags/admin/tags.admin.php* – редактирование тегов;
4. *modules/tags/admin/tags.admin.action.php* – обработка POST-запросов при работе с тегами в административной части;
5. *modules/tags/admin/tags.admin.config.php* – настройки модуля;
6. *modules/tags/admin/tags.admin.inc.php* – подключение модуля к административной части других модулей;
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.rows.php* – шаблон списка элементов, к которым

прикреплен тег;

16. *modules/tags/views/tags.view.rows_block.php* – шаблон облака тегов;

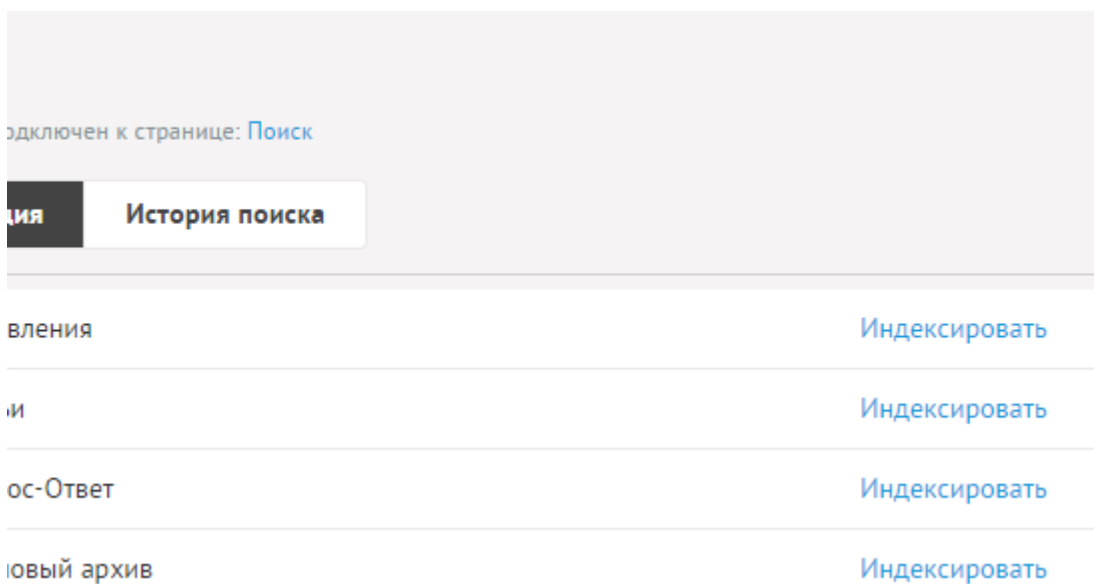
17. *modules/tags/views/tags.view.show_block.php* – шаблон облака тегов.

Поиск

- [Индексация](#)
- [Подключение](#)
- [История поиска](#)
- [Настройки](#)
- [Шаблонные теги](#)
- [База данных](#)
- [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 изменить `modules/модуль/модуль.search.php`:

```
class Модуль_search_config
{
    public $config = array();
}
```

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', ...),
            'rating' => рейтинг_модуля_в_поисковой_выдаче
        ),
        'таблица_2' => array(
            ...
        ),
        ...
    );
}
```

Можно задать поле "param", тогда будут проиндексированы все значения дополнительных характеристик для элементов модуля. Например, дополнительные характеристики товаров в магазине.

В поисковой выдаче на сайте результаты поиска оформляются так же, как в списке элементов этого модуля. Для этого данные обрабатываются функцией `elements()` из модели модуля *modules/модуль/модуль.model.php* и выводятся шаблоном *modules/модуль/views/модуль.view.list.php*.

Можно задать свою обработку данных для поиска. Для этого нужно определить функцию `search()` в файле *modules/модуль/модуль.model.php*.

Пример:

Так обработка данных о найденных товарах идет через функцию `search()` в файле *modules/shop/shop.model.php*.

А обработка данных о найденных новостях идет через функцию `elements()` в файле *modules/news/news.model.php*, так как функцию `search()` в этом файле не описана.

Шаблон также можно применить отличный от шаблона по умолчанию. Для этого достаточно создать файл *modules/модуль/views/модуль.view.list_search.php*.

Пример:

Найденные товары оформляются в результатах поиска шаблоном *modules/shop/views/shop.view.list_search.php*.

А найденные новости оформляются также, как в списке новостей шаблоном *modules/news/views/news.view.list.php*.

Описанные выше особенности относятся только к элементам модулей (новости, товары, статьи и пр.). Категории и производители в модуле, а также страницы сайта выводятся стандартным шаблоном *modules/search/views/search.view.show.php*.

Подключение

Подключаемая часть – файл *modules/search/search.inc.php*. В нем описан класс *Search_inc*. В модуле к объекту класса можно обратиться через переменную `$this->diafan->_search`. Экземпляр класса создается при первом вызове переменной.

Методы

`void index_all ()` – Индексирует весь сайт.

Пример:

```
// индексирует все страницы на сайте, которые определены для индексации в файле настроек
$this->diafan->_search->index_all();
```

`void index_module (string $module_name)` – Индексирует модуль.

- string \$module_name: название модуля

Пример:

```
// индексирует модуль «Интернет магазин»
$this->diafan->_search->index_module('shop');
```

`void index_site_module (array $site, [boolean $out = true], [integer $count_sites = 0])` – Индексирует страницы модуля, прикрепленного к странице сайта.

- array \$site: данные о странице сайта
- boolean \$out: функция используется как внешняя
- integer \$count_sites: количество страниц, к которым прикреплен модуль

Пример:

```
$site = array(
    "id" => 13,
    "module_name" => "shop",
);
// индексирует страницу, с прикрепленным модулем
$this->diafan->_search->index_site_module($site);
```

`void index_elements (array $rows, string $table_name)` – Индексирует группу элементов.

- array \$rows: массив данных об индексируемых элементах
- string \$table_name: таблица элементов

Пример:

```

$rows = array(
    array(
        "id" => 1,
        "search_no_show" => 0,
        "act1" = 1,
        "act2" => 0,
        "access" => 0,
        "name" => "Газонокосилка электрическая",
        "anons" => "Хорошая газонокосилка по приемлемой цене.",
        "text" => "Срок службы обычно 3-4 года, регулируемая высота покоса, длинный
шнур.",
        "article" => "K1200"
    ),
    array(
        "id" => 2,
        "search_no_show" => 0,
        "act1" = 1,
        "act2" => 1,
        "access" => 0,
        "name" => "Газонокосилка бензиновая",
        "name2" => "Lawn mowers",
        "anons" => "Лучшая газонокосилка, но дороже.",
        "anons2" => "Best lawn mower, but more expensive.",
        "text" => "Срок службы обычно 5-6 лет, нужно заправлять бензином, регулируемая
высота покоса, длинный шнур.",
        "text2" => "Life is normally 5-6 years, you need to fill petrol, height adjustable mowing,
long cord.",
        "article" => "K1200"
    ),
);
// индексирует несколько товаров
$this->diafan->_search->index_elements($rows, "shop");

```

void **index_element** (array \$row, strign \$table_name) – Индексирует один элемент.

- array \$row: данные об индексируемом элементе
- strign \$table_name: таблица элемента

Пример:

```

$row = array(
    "id" => 1,
    "search_no_show" => 0,
    "act1" = 1,
    "act2" => 0,
    "access" => 0,
    "name" => "Газонокосилка электрическая",
    "anons" => "Хорошая газонокосилка по приемлемой цене.",

```

```

    "text" => "Срок службы обычно 3-4 года, регулируемая высота покоса, длинный шнур.",
    "article" => "K1200"
);
// индексирует товар
$this->diafan->_search->index_element($row, "shop");

```

void delete (integer|array \$element_ids, string \$table_name) – Удаляет один или несколько элементов.

- integer|array \$element_ids: номер одного или нескольких элементов
- string \$table_name: таблица элемента

Пример:

```

// удаляет производителя товаров из индекса
$this->diafan->_search->delete(2, "shop", "brand");

```

void delete_module (string \$module_name) – Удаляет весь индекс модуля.

- string \$module_name: название модуля

Пример:

```

// удаляет из поискового индекса новости и категории новостей
$this->diafan->_search->delete_module("news");

```

void delete_sites (array \$site_ids) – Удаляет индекс модулей, прикрепленных к страницам сайта.

- array \$site_ids: идентификаторы страниц сайта

Пример:

```

// удаляет индекс модулей, прикрепленных к страницам сайта ID 3, 4, 6
$this->diafan->_search->delete_sites(array(3, 4, 6));

```

array|boolean false check_module_config (string \$module_name, [string \$table_name = '']) – Проверяет есть ли настройки поиска для модуля и таблицы модуля.

- string \$module_name: модуль
- string \$table_name: таблица

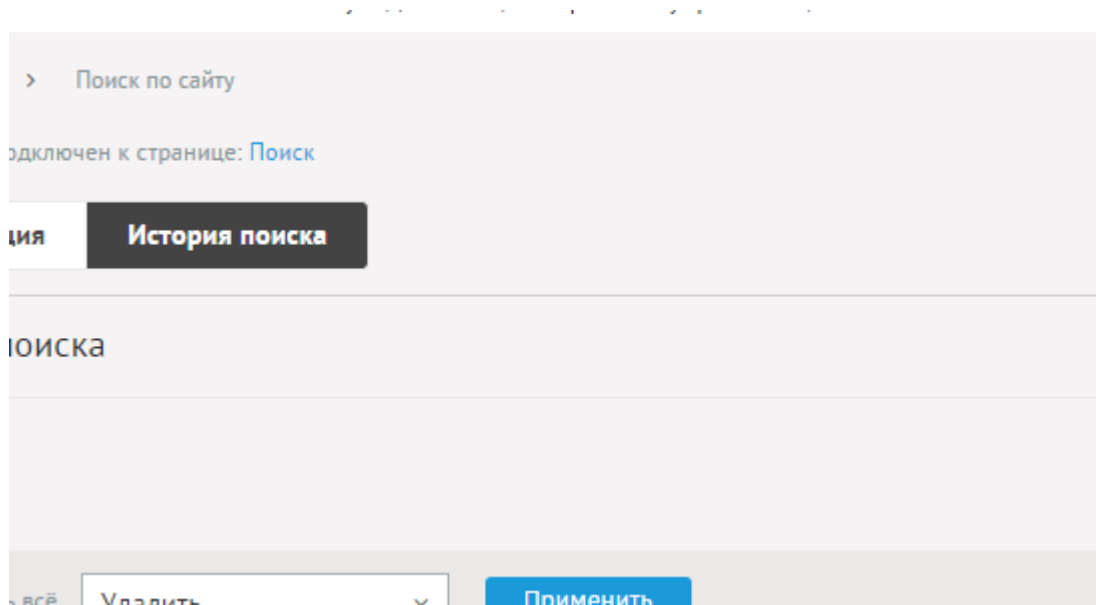
Пример:

```

$config = $this->diafan->_search->check_module_config("shop", "shop_brand");
if($config)
{
    // индексируем производителей в магазине
}

```

История поиска



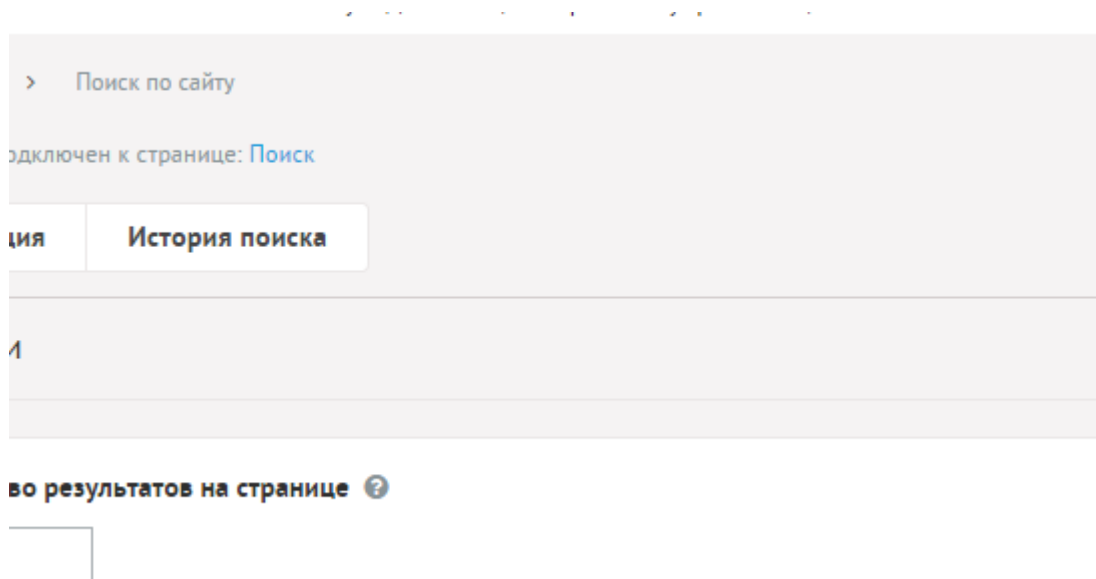
Поисковые запросы имеют следующие характеристики:

Основные:

- **Запрос.**
- **Дата.**

Список последних поисковых запросов. Количество запросов в истории поиска определяется в настройках модуля.

Настройки



- **Индексировать автоматически** – Если не отмечено, индексация происходит только в административном интерфейсе «Поиск по сайту».
- **Количество результатов на странице** – Количество одновременно выводимых результатов поиска в списке.
- **Включить «Показать ещё»** – На странице результатов появится кнопка «Показать

ещё». Увеличивает количество одновременно выводимых результатов в списке.

- **Количество последних запросов в истории поиска.**
- **Искать все слова сразу** – Если не отмечено, ищет хотя бы одно слово.
- **Искать часть слова** – Если не отмечено, ищет слово и словоформы целиком. Если отмечено, то может найти длинное слово уже по первым трем-четырем символам. Если отмечено, то настройка «Искать все слова сразу» не работает.
- **Минимальная длина слова** – При изменении параметра необходимо переиндексировать весь сайт. Чем меньше этот параметр, тем больше места занимает поисковый индекс. Значение по умолчанию: 3.

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»;
- **ajax** – подгружать результаты поиска без перезагрузки страницы.: **true** – результаты поиска подгружаются, по умолчанию будет перезагружена вся страница.;
- **defer** – маркер отложенной загрузки шаблонного тега: **event** – загрузка контента только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** – загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка контента только по желанию пользователя;
- **defer_title** – текстовая строка, выводимая на месте появления загружаемого контента с помощью отложенной загрузки шаблонного тега;
- **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

В шаблоне тега можно получить значение любого атрибута через переменную `$result["attributes"]["название атрибута"]`.

База данных

[{search_history}](#) – История поисковых запросов

[{search_index}](#) – Связи слов и проиндексированных для поиска элементов

[{search_keywords}](#) – Индексированные для поиска слова

[{search_results}](#) – Индексированные для поиска элементы

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/js/search.show_search_top.js* – JS-сценарий формы поиска;
6. *modules/search/search.php* – контроллер;
7. *modules/search/search.action.php* – обрабатывает полученные данные из формы;
8. *modules/search/search.history.php* – экспорт истории поиска;
9. *modules/search/search.inc.php* – подключение модуля;
10. *modules/search/search.install.php* – установка модуля;
11. *modules/search/search.model.php* – модель;
12. *modules/search/views/search.view.rows.php* – шаблон результатов поиска по сайту;
13. *modules/search/views/search.view.show.php* – шаблон результатов поиска по сайту;
14. *modules/search/views/search.view.show_search.php* – шаблон формы поиска по сайту;
15. *modules/search/views/search.view.show_search_404.php* – шаблон формы поиска по сайту, `template=404`;
16. *modules/search/views/search.view.show_search_top.php* – шаблон формы поиска по сайту, `template=top`.

Вопрос-Ответ

- [Вопросы - ответы](#)
- [Категории](#)
- [Статистика](#)
- [Настройки](#)
- [Шаблонные теги](#)
- [База данных](#)
- [Файлы](#)

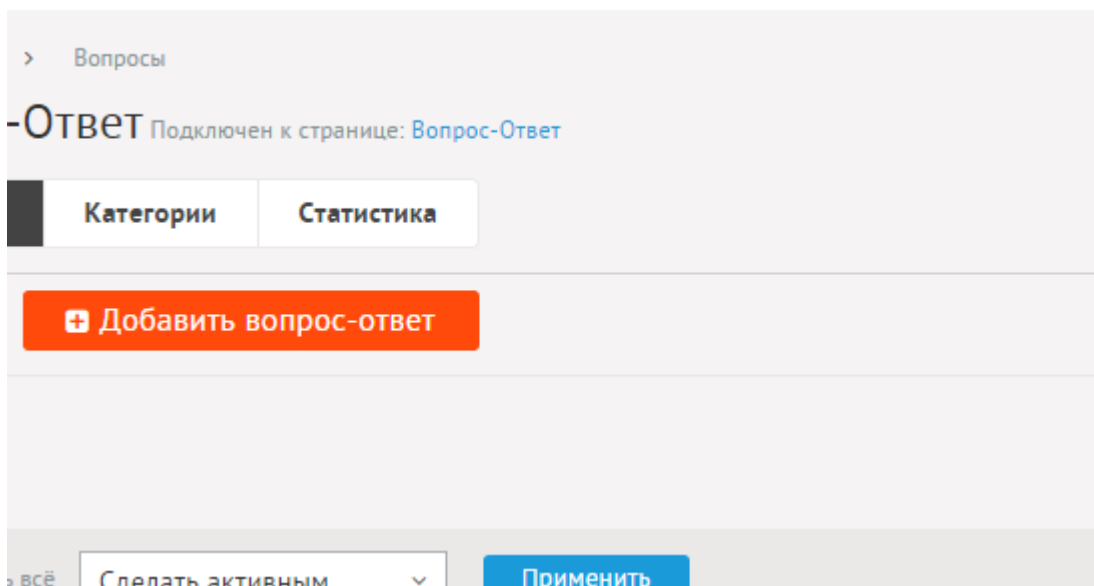
Модуль позволяет пользователям задавать вопросы на сайте, а администратору публиковать ответы на эти вопросы с возможностью отправки ответа на e-mail пользователя, задавшего вопрос.

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

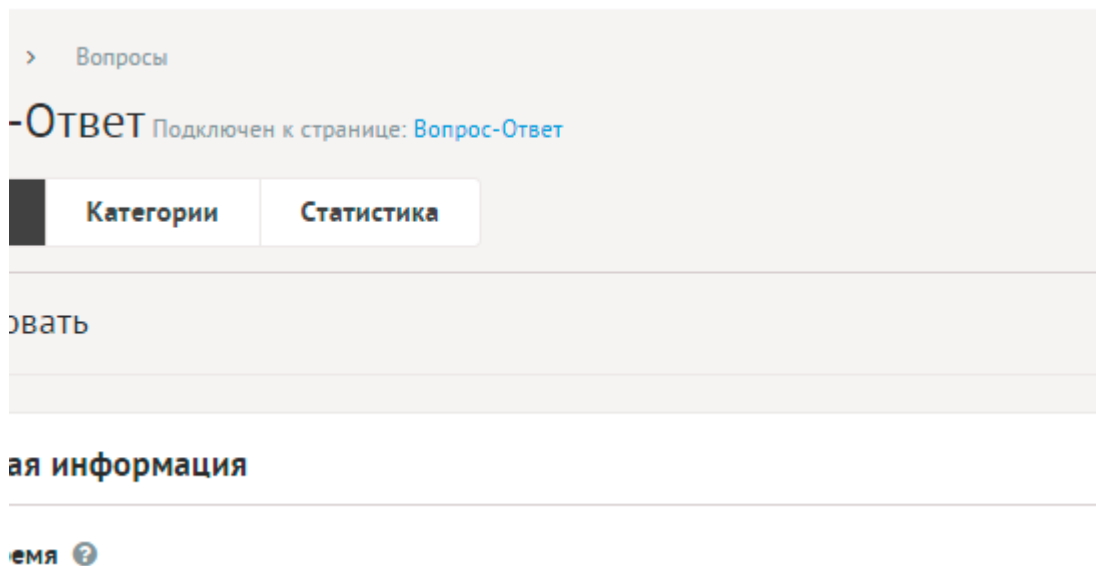
Модуль «Вопрос-Ответ» можно прикреплять к различным страницам сайта. Если модуль прикреплен к нескольким страницам, в административной части модуля появляется специальный фильтр «Раздел сайта» с возможностью фильтрации вывода (списка вопросов, категорий и настроек) по принадлежности к странице.

Модуль состоит из двух частей: категории вопросов и вопросы-ответы.

Вопросы - ответы



В списке вопросы выводятся с приоритетом новизны.



Вопросы имеют следующие характеристики.

Основные:

- **Дата и время** – Дата и время поступления вопроса в формате дд.мм.гггг чч:мм, приходит с сайта.
- **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.
- **Отправить ответ на e-mail** – 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.
- **Ответ** – Текстовое поле для ответа. Параметр имеет разные значения для разных языковых версий сайта.
- **Часто задаваемый вопрос** – Используется в шаблонной функции show_block.
- **Динамические блоки.**
- **Вопрос задал.**
- **Автор** – Пользователь, создавший вопрос в форме на сайте.
- **Имя** – Имя отправителя вопроса. Параметр имеет разные значения для разных языковых версий сайта.
- **Email** – Электронный ящик получателя ответа.
- **Прикрепленные файлы.**
- **Категория** – Категория, к которой относится вопрос. Список категорий редактируется во вкладке выше. 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».
- **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».

- **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.
- **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».

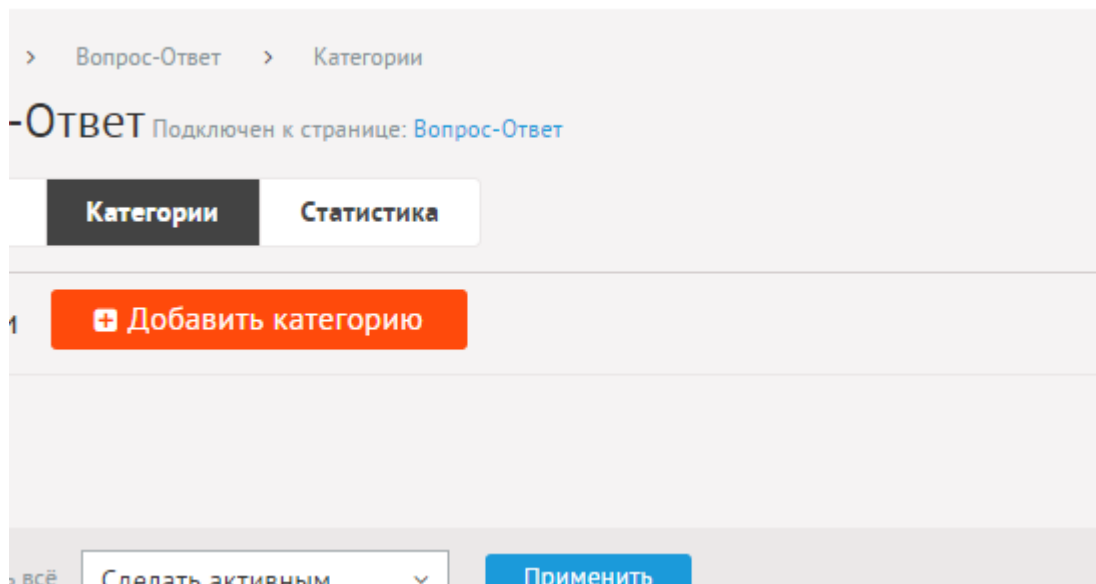
Дополнительные:

- **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*.
- **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.
- **Параметры SEO.**
- **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.
- **Канонический тег** – URL канонической страницы вида: <http://site.ru/pseudossylka/>, на которую переносится "ссылочный вес" данной страницы. Используется для страниц с похожим или дублирующимся контентом (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.

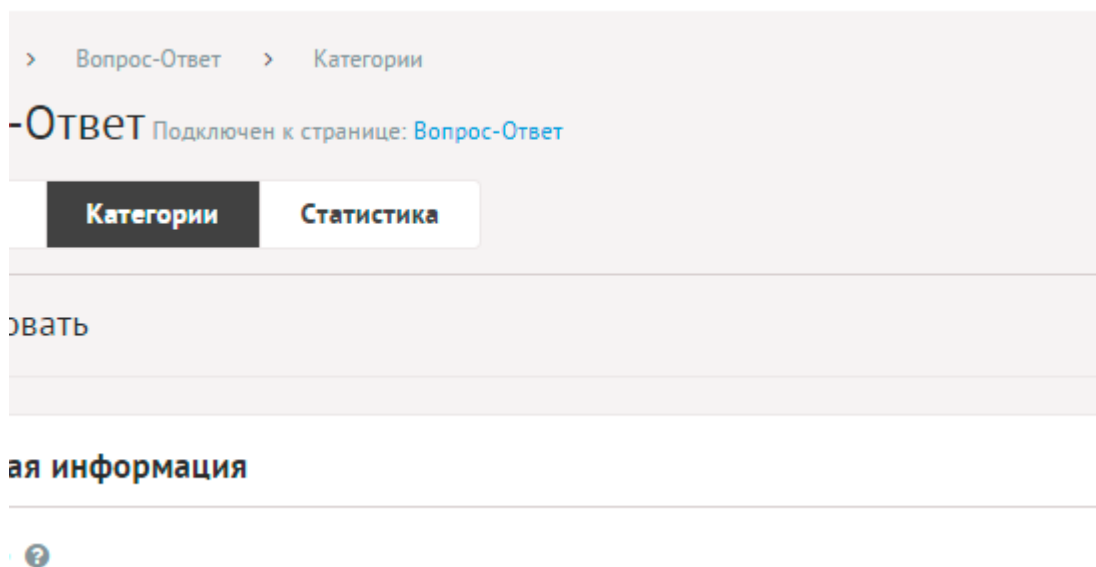
- **Не индексировать** – Запрет индексации текущей страницы, если отметить, у страницы выведется тег: `<meta name="robots" content="noindex">` (SEO-специалисту).
- **Changefreq** – Вероятная частота изменения этой страницы. Это значение используется для генерирования файла *sitemap.xml*. Подробнее читайте 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).
- **Параметры показа.**
- **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»](#).
- **Оформление.**
- **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!).
- **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*.

Категории



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

При удалении категории удаляются все вложенные подкатегории и вопросы.



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.
- **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».
- **Динамические блоки.**
- **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»](#). Опция «HTML-код» позволяет отключить визуальный редактор для текущего поля. Значение этой настройки будет учитываться и при последующем редактировании. 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»](#). Опция «HTML-код» позволяет отключить визуальный редактор для текущего поля. Значение этой настройки будет учитываться и при последующем редактировании. 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*.

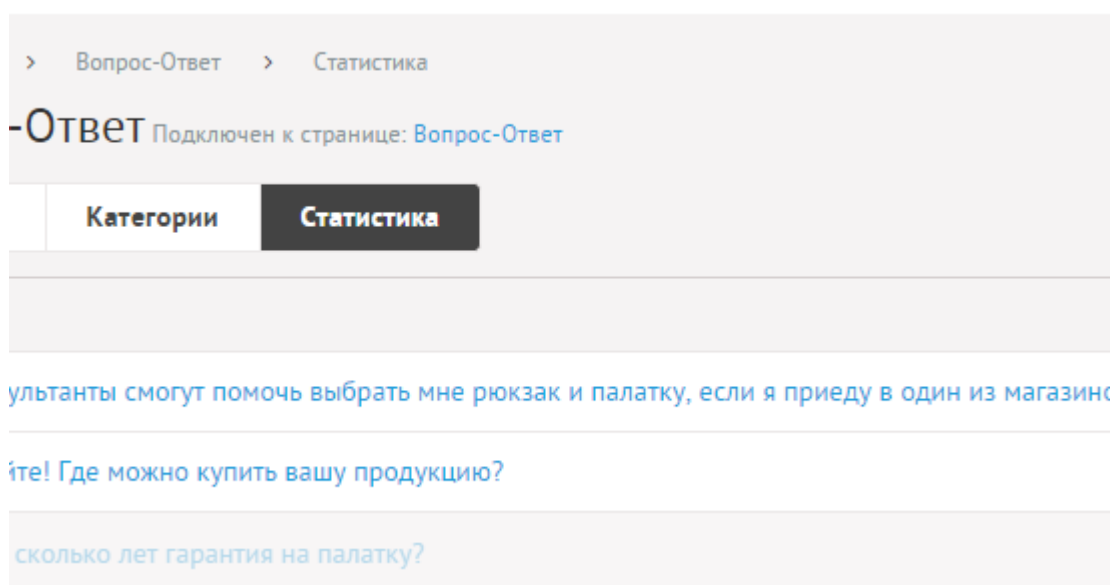
Дополнительные:

- **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*.
- **Параметры SEO.**
- **Заголовок окна в браузере, тег Title** – Если не заполнен, тег *Title* будет автоматически сформирован как «Название категории – 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, 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.
- **Канонический тег** – URL канонической страницы вида: <http://site.ru/pseudossylka/>, на которую переносится "ссылочный вес" данной страницы. Используется для страниц с похожим или дублирующимся контентом (SEO-специалисту). Параметр имеет разные

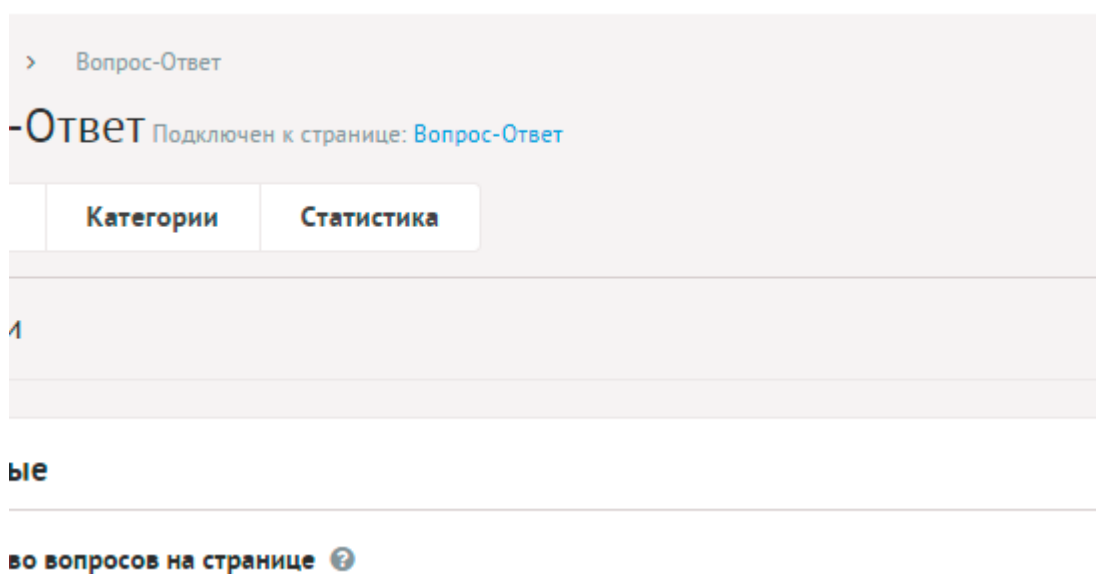
значения для разных языковых версий сайта.

- **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.
- **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).
- **Параметры показа.**
- **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»](#).
- **Шаблоны.**
- **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 списка элементов** – Шаблон вывода контента модуля на странице элементов списка в категории (веб-мастеру и программисту, не меняйте этот параметр, если не уверены в результате!). Значение параметра важно для AJAX.
- **Шаблон страницы элемента** – 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!).

Статистика



Настройки



- **Основные.**
- **Количество вопросов на странице** – Количество одновременно выводимых вопросов в списке.
- **Включить «Показать ещё»** – На странице вопросов появится кнопка «Показать ещё». Увеличивает количество одновременно выводимых вопросов в списке.
- **Количество символов для сокращения вопроса и ответа в списке** – Если не задано, вопрос и ответ не будут сокращаться.
- **Формат даты** – Позволяет настроить отображение даты в модуле. Возможные значения:
 - 01.05.2016,
 - 01.05.2016 14:45,
 - 1 мая 2016 г.,
 - 1 мая,

- 1 мая 2016, понедельник,
- вчера 15:30,
- не отображать.
- **Только зарегистрированные пользователи могут задавать вопросы** – Вопросы могут добавлять только зарегистрированные пользователи.
- **Открывать вопрос на отдельной странице** – Если не отмечена, вопросы будут одним списком без возможности открыть отдельную страницу.
- **Использовать защитный код (капчу)** – Для добавления вопроса пользователь должен ввести защитный код.
- **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.
- **Категории.**
- **Использовать категории** – Позволяет включить/отключить категории вопросов.
- **Количество категорий на странице** – Количество одновременно выводимых категорий в списке на первой страницы модуля. Параметр появляется, если отмечена опция «Использовать категории».
- **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».
- **Уведомления.**
- **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><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.
- **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.

- **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 вопросов на e-mail** – Возможность уведомления администратора о поступлении новых вопросов 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

вопросов 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 вопросов на e-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 – e-mail пользователя,
- %files – название прикрепляемых файлов.

Параметр появляется, если отмечена опция «Уведомлять о поступлении новых вопросов на e-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».
- **Подключения.**
- **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».
- **Подключить комментарии к вопросам** – Подключение модуля «Комментарии». Параметр не будет включен, если модуль «Комментарии» не установлен. Подробности см. в разделе [модуль «Комментарии»](#).
- **Подключить 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.
- **Автоформирование для SEO.**
- **Шаблон для автоматического генерирования Title** – Если шаблон задан и для вопроса не прописан заголовок *Title*, то заголовок автоматически генерируется по шаблону. В шаблон можно добавить:
 - %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 и для категории не прописан заголовок *Title*, то заголовок автоматически генерируется по шаблону. В шаблон можно добавить:
 - %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 и для вопроса не заполнено поле *Keywords*, то поле *Keywords* автоматически генерируется по шаблону. В шаблон можно добавить:
 - %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 и для категории не заполнено поле *Keywords*, то поле *Keywords* автоматически генерируется по шаблону. В шаблон можно добавить:
 - %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 и для вопроса не заполнено поле *Description*, то поле *Description* автоматически генерируется по шаблону. В шаблон можно добавить:
 - %name – название,
 - %category – название категории,
 - %parent_category – название категории верхнего уровня,
 - %anons – краткое описание (SEO-специалисту).

Параметр имеет разные значения для разных языковых версий сайта.

- **Шаблон для автоматического генерирования Description для категории** – Если шаблон задан и для категории не заполнено поле *Description*, то поле *Description* автоматически генерируется по шаблону. В шаблон можно добавить:
 - %name – название категории,
 - %parent – название категории верхнего уровня,
 - %anons – краткое описание (SEO-специалисту).

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

- **Оформление.**
- **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 элементов в списке** – По умолчанию *modules/faq/views/faq.view.rows.php*. Параметр для разработчиков! Не устанавливайте, если не уверены в результате. Значение параметра важно для AJAX.
- **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 элементов в списке 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. Значение параметра важно для AJAX.
- **Шаблон для страницы элемента** – По умолчанию, *modules/faq/views/faq.view.id.php*. Параметр для разработчиков! Не устанавливайте, если не уверены в результате.
- **Дополнительно.**
- **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. Можно указать отрицательное значение, тогда будут исключены вопросы из указанного раздела. По умолчанию выбираются все страницы;
- **cat_id** – категории вопросов, если в настройках модуля отмечено «Использовать категории». Идентификаторы категорий перечисляются через запятую. Можно указать отрицательное значение, тогда будут исключены вопросы из указанной категории. Можно указать значение **current**, тогда будут показаны вопросы из текущей (открытой) категории или из всех категорий, если ни одна категория не открыта. По умолчанию категория не учитывается, выводятся все вопросы;
- **sort** – сортировка вопросов: **date** – по дате (по умолчанию), **rand** – в случайном порядке;
- **often** – часто задаваемые вопросы : **true** – выводятся только вопросы с пометкой «Часто задаваемый вопрос», по умолчанию пометка «Часто задаваемый вопрос» игнорируется;
- **only_module** – выводить блок только на странице, к которой прикреплен модуль «Вопрос-Ответ»: **true** – выводить блок только на странице модуля, по умолчанию блок будет выводиться на всех страницах;
- **tag** – тег, прикрепленный к вопросам;
- **defer** – маркер отложенной загрузки шаблонного тега: **event** – загрузка контента только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** – загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка контента только по желанию пользователя;
- **defer_title** – текстовая строка, выводимая на месте появления загружаемого контента с помощью отложенной загрузки шаблонного тега;
- **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

В шаблоне тега можно получить значение любого атрибута через переменную `$result["attributes"]["название атрибута"]`.

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);
- **defer** – маркер отложенной загрузки шаблонного тега: **event** – загрузка контента только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** – загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка контента только по желанию пользователя;
- **defer_title** – текстовая строка, выводимая на месте появления загружаемого контента с помощью отложенной загрузки шаблонного тега;
- **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

В шаблоне тега можно получить значение любого атрибута через переменную `$result["attributes"]["название атрибута"]`.

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»;
- **only_module** – выводить форму только на странице, к которой прикреплен модуль «Вопрос-Ответ»: **true** – выводить форму только на странице модуля, по умолчанию форма будет выводиться на всех страницах;
- **defer** – маркер отложенной загрузки шаблонного тега: **event** – загрузка контента только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** – загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка контента только по желанию пользователя;
- **defer_title** – текстовая строка, выводимая на месте появления загружаемого контента с помощью отложенной загрузки шаблонного тега;
- **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

В шаблоне тега можно получить значение любого атрибута через переменную `$result["attributes"]["название атрибута"]`.

База данных

[{faq}](#) – Вопросы и ответы

[{faq_category}](#) – Категории вопросов и ответов

[{faq_category_parents}](#) – Родительские связи категорий вопросов и ответов

[{faq_category_rel}](#) – Связи вопросов и ответов с категориями

[{faq_counter}](#) – Счетчик вопросов и ответов

[{faq_rel}](#) – Связи похожих вопросов и ответов

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.dashboard.php* – вопросы-ответы для событий;
7. *modules/faq/admin/faq.admin.menu.php* – карта ссылок для модуля «Меню на сайте»;
8. *modules/faq/faq.php* – контроллер;
9. *modules/faq/faq.action.php* – обработка запроса при отправке сообщения из формы;
10. *modules/faq/faq.install.php* – установка модуля;
11. *modules/faq/faq.model.php* – модель;
12. *modules/faq/faq.search.php* – настройки для поисковой индексации для модуля «Поиск»;
13. *modules/faq/faq.sitemap.php* – карта ссылок для модуля «Карта сайта»;
14. *modules/faq/views/faq.view.first_page.php* – шаблон первой страницы модуля, если в настройках модуля подключен параметр «Использовать категории»;
15. *modules/faq/views/faq.view.form.php* – шаблон формы добавления вопроса;
16. *modules/faq/views/faq.view.id.php* – шаблон страницы вопроса;
17. *modules/faq/views/faq.view.list.php* – шаблон списка вопросов и ответов;
18. *modules/faq/views/faq.view.rows.php* – шаблон элементов в списке вопросов и ответов;
19. *modules/faq/views/faq.view.rows_block.php* – шаблон блока вопросов и ответов;
20. *modules/faq/views/faq.view.rows_block_rel.php* – шаблон блока похожих вопросов и ответов;
21. *modules/faq/views/faq.view.show_block.php* – шаблон блока вопросов и ответов;
22. *modules/faq/views/faq.view.show_block_rel.php* – шаблон блока похожих вопросов и ответов.

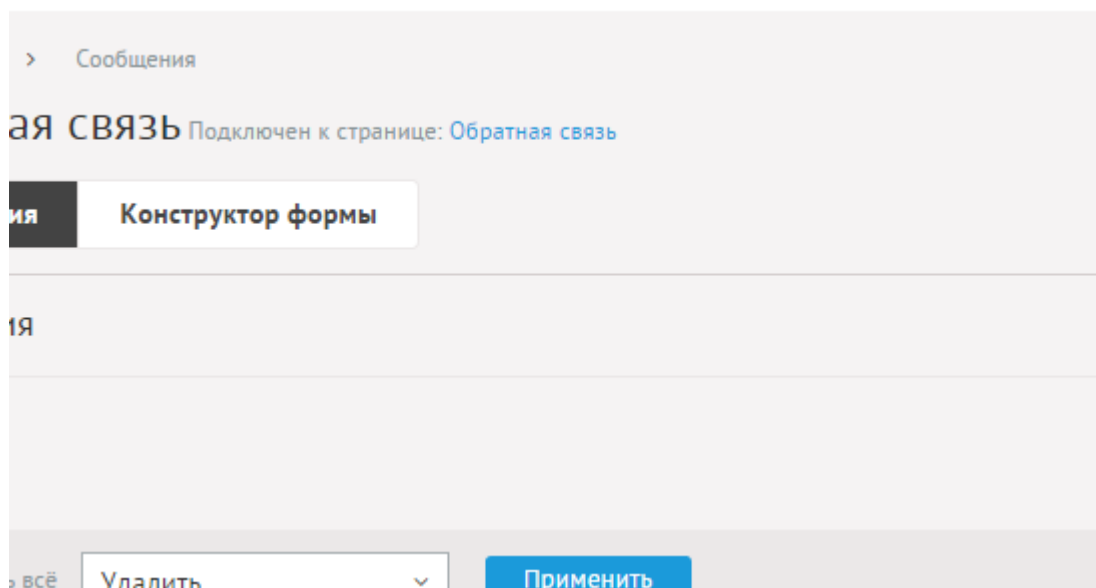
Обратная связь

- [Сообщения](#)
- [Конструктор формы](#)
- [Настройки](#)
- [Шаблонные теги](#)
- [База данных](#)
- [Файлы](#)

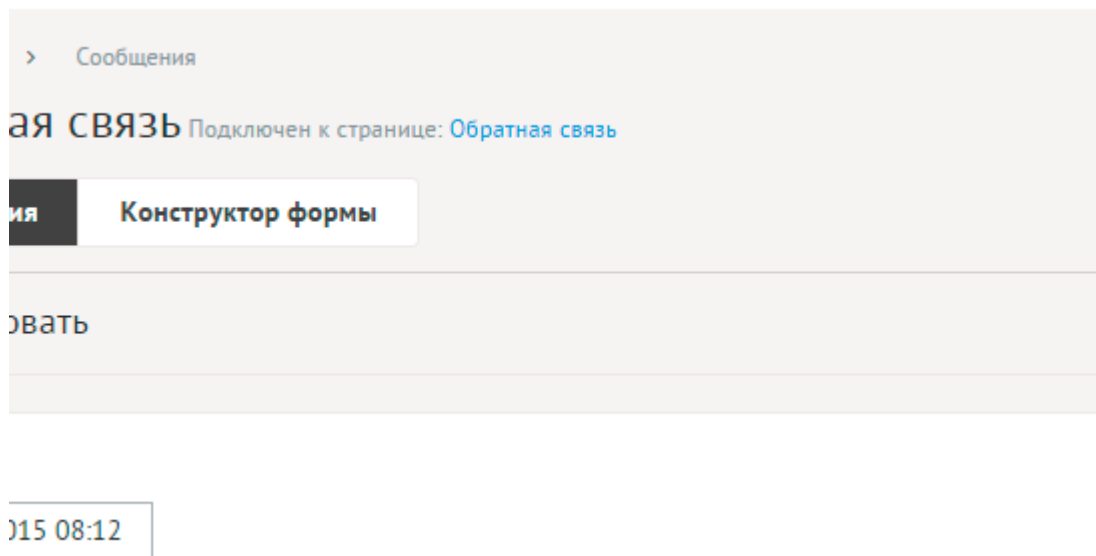
Модуль позволяет пользователям оставлять сообщения на сайте и отвечать отправителям в административной части модуля. Модуль относится к части CMS «Интерактив» и может быть установлен опционально.

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



В списке сообщения сортируются в порядке добавления пользователями.



Модуль имеет следующие характеристики.

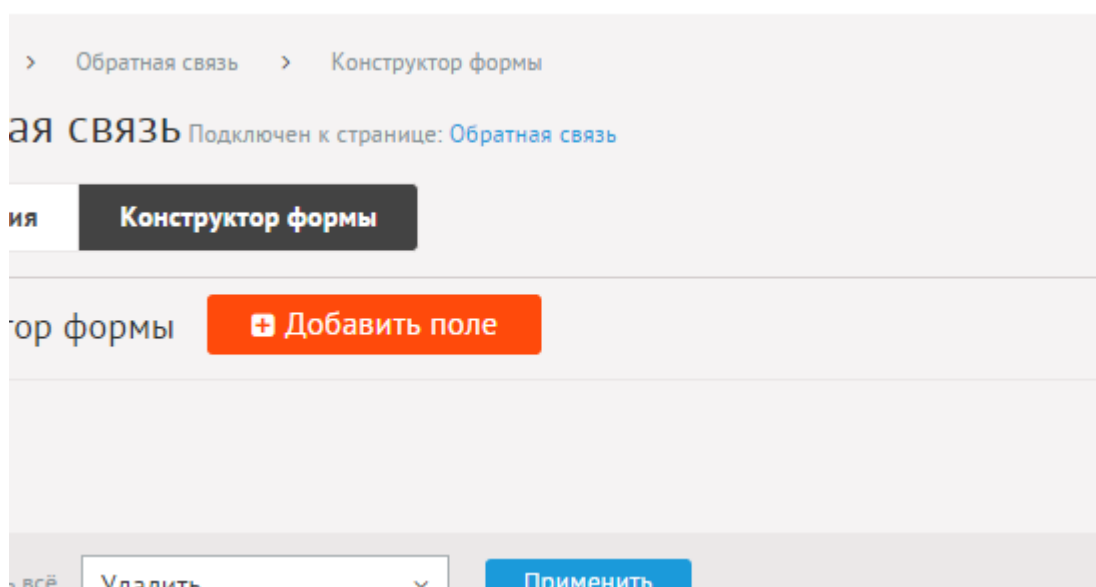
Основные:

- **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.
- **Страница, с которой отправлено сообщение** – Полный адрес страницы, с которой пользователь отправил 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 настройках модуля. Письмо не может быть отправлено, если не заполнено текстовое поле для ответа или e-mail получателя. Поле «Отправить ответ» появляется, если в конструкторе формы есть 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.

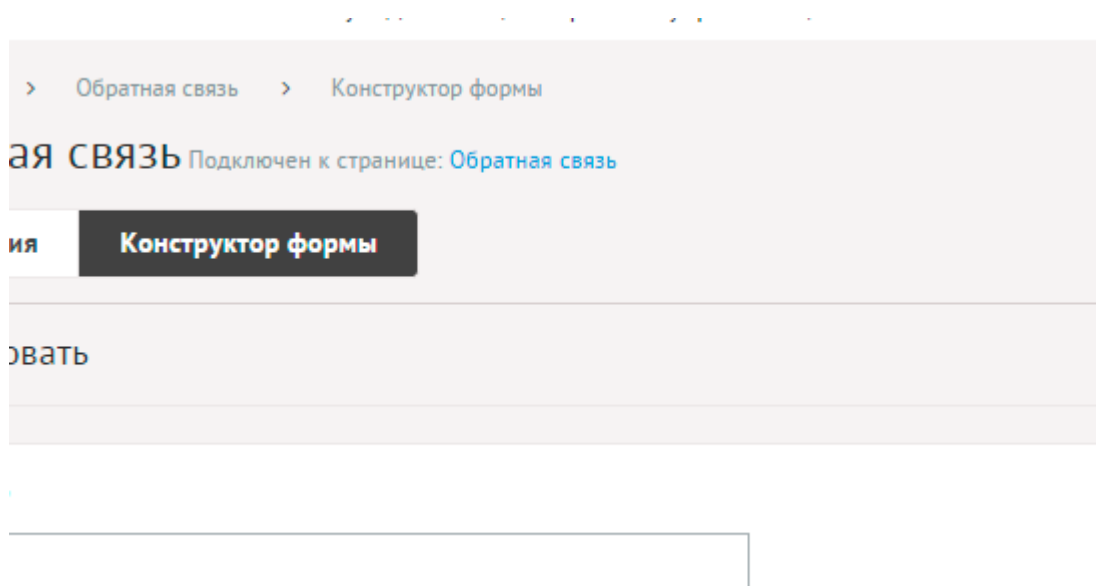
Если есть дополнительно поле с типом электронный ящик, то в административной части можно ответить на сообщение, отправив письмо на указанный ящик. Тема письма, содержание и обратный адрес настраиваются в конфигурации модуля.

Конструктор формы



На разных страницах сайта структура формы различная. Это обеспечивает конструктор формы.

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,
 - переключатель,
 - выпадающий список,
 - список с выбором нескольких значений,
 - электронный ящик,
 - телефон,
 - 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

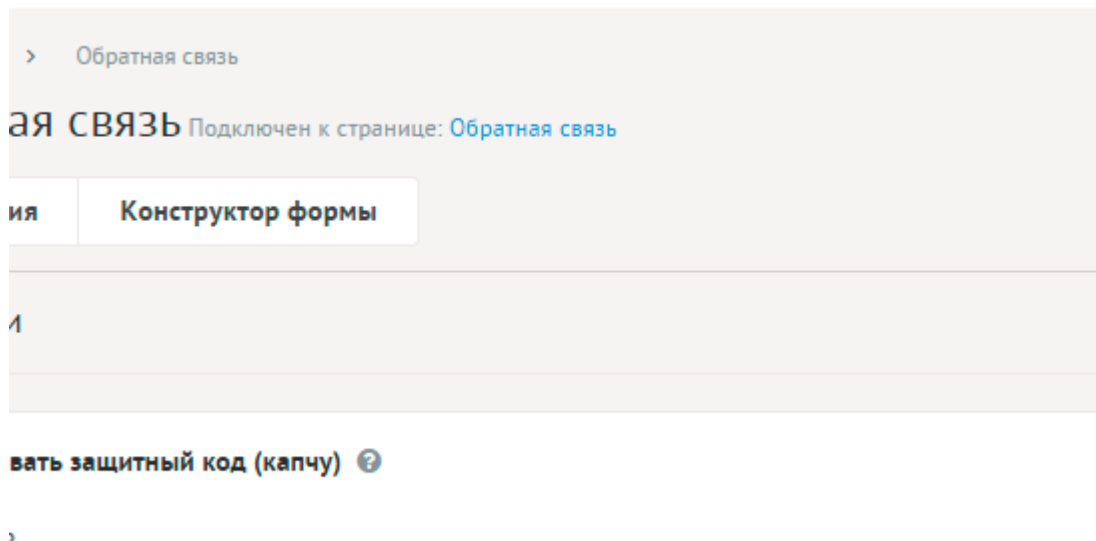
- **Распознавать изображения** – Позволяет прикрепленные файлы в формате JPEG, GIF, PNG отображать как изображения. Параметр выводится, если тип характеристики задан как «файлы».
- **Размер для большого изображения** – Размер изображения, отображаемый в пользовательской части сайта при увеличении изображения 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.

Настройки



- **Использовать защитный код (капчу)** – Для добавления сообщения пользователь должен ввести защитный код.
- **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></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.

- **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 сообщений на e-mail** – Возможность уведомления администратора о поступлении новых сообщений 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 сообщений 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 сообщений на e-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 e-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;
- **defer** – маркер отложенной загрузки шаблонного тега: **event** – загрузка контента только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** – загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка контента только по желанию пользователя;
- **defer_title** – текстовая строка, выводимая на месте появления загружаемого контента с помощью отложенной загрузки шаблонного тега;
- **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

В шаблоне тега можно получить значение любого атрибута через переменную `$result["attributes"][$"название атрибута"]`.

База данных

[{feedback}](#) – Сообщения из обратной связи

[{feedback_param}](#) – Поля конструктора формы обратной связи

[{feedback_param_element}](#) – Значения полей конструктора формы обратной связи, заполненные в сообщении

[{feedback_param_select}](#) – Варианты значений полей конструктора с типом список

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.dashboard.php* – сообщения обратной связи для событий;
5. *modules/feedback/admin/feedback.admin.param.php* – конструктор формы обратной связи;
6. *modules/feedback/feedback.php* – контроллер;
7. *modules/feedback/feedback.action.php* – обработка POST-запроса;
8. *modules/feedback/feedback.install.php* – установка модуля;
9. *modules/feedback/feedback.model.php* – модель;
10. *modules/feedback/views/feedback.view.form.php* – шаблон формы добавления сообщения в обратной связи;
11. *modules/feedback/views/feedback.view.images.php* – шаблон вывода прикрепленных к сообщению изображений.

Форум

- [Блоки](#)
- [Категории](#)
- [Темы](#)
- [Сообщения](#)
- [Настройки](#)
- [Шаблонные теги](#)
- [База данных](#)
- [Файлы](#)

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

Полная административная часть модуля состоит из четырех частей:

- блоки
- категории
- темы
- сообщения

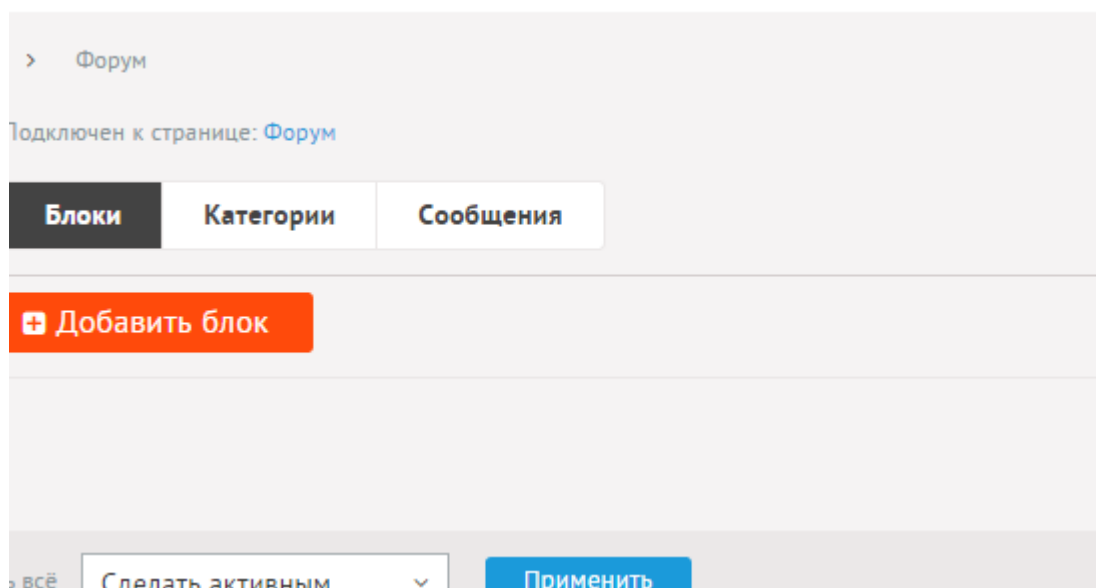
Первые две – блоки и категории – определяют структуру форума. Поэтому администратор должен их описать в самом начале, до запуска форума.

Темы и сообщения форума создают пользователи, в соответствии с настройками модуля.

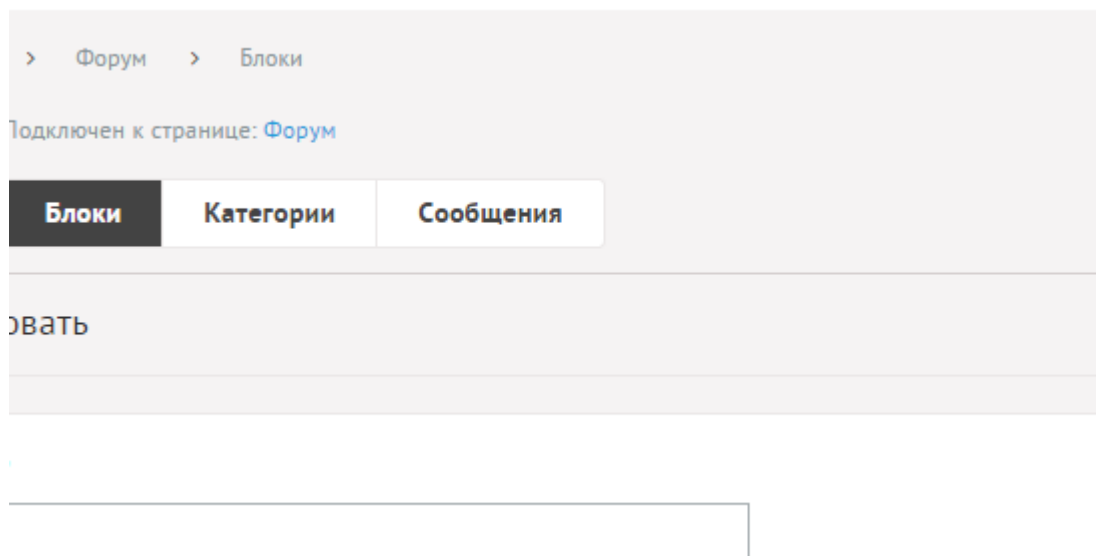
Некоторых пользователей можно назначить модераторами - на вкладке «Права для пользовательской части» (*Пользователи сайта – Права доступа*). Модераторы получают права на удаление, блокировку и редактирование тем и сообщений.

Обратите внимание: контент модуля «Форум» не делится по разделам. Это значит, что прикрепление модуля к разным страницам сайта приведет к полному дублированию блоков, категорий, тем и сообщений форума.

Блоки



Блоки – глобальное разделение категорий на логические части. Блоки не имеют отдельных страниц в пользовательской части. Они отображаются только на первой странице модуля.



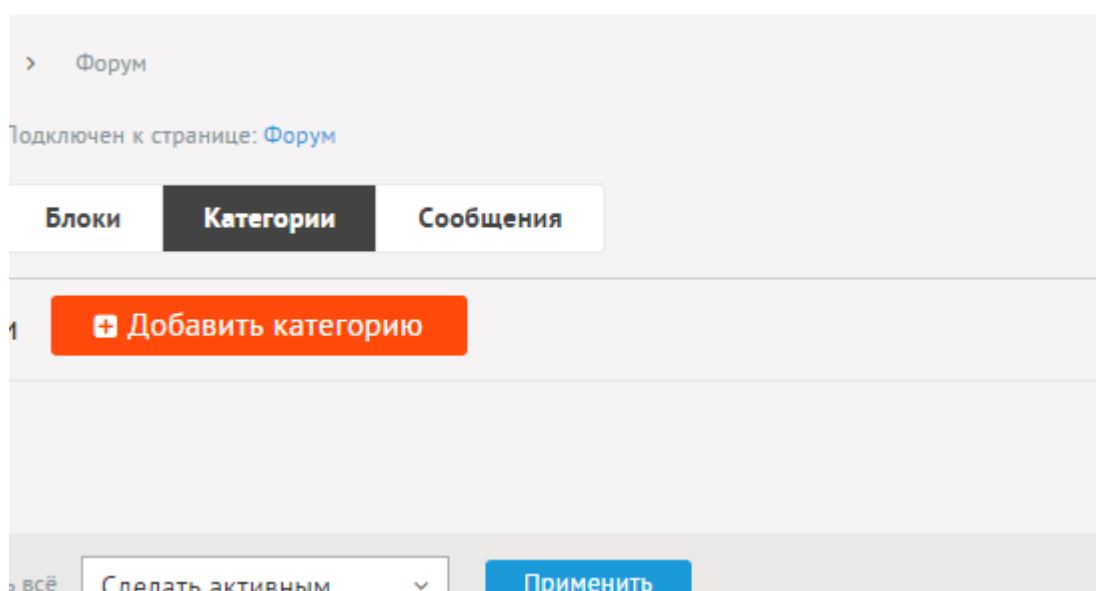
При редактировании для блоков можно задать следующие поля.

Основные:

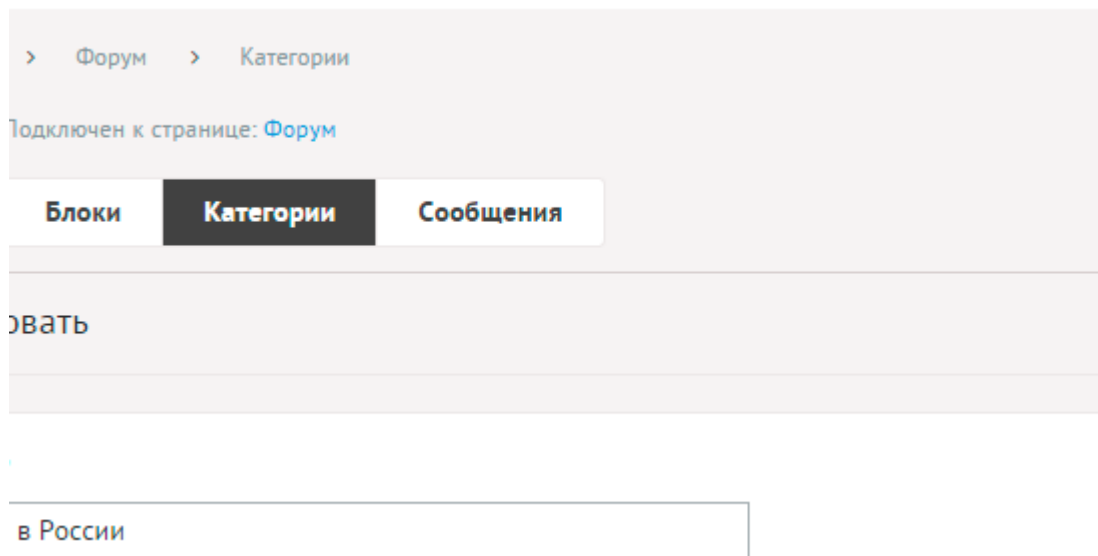
- **Название.**
- **Опубликовать на сайте.**
- **Сортировка: установить перед** – Редактирование порядка следования блока в списке. Поле доступно для редактирования только для блоков, отображаемых на сайте.

Категории

Категории – это более детальное разделение тем. Категории имеют отдельные страницы в пользовательской части



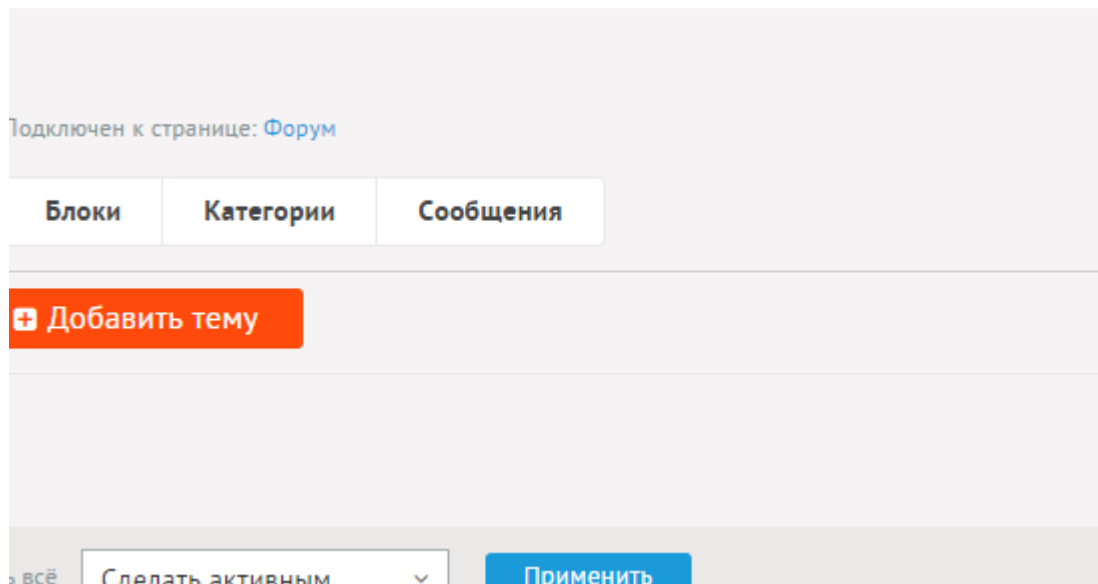
При редактировании для категорий можно задать следующие поля.



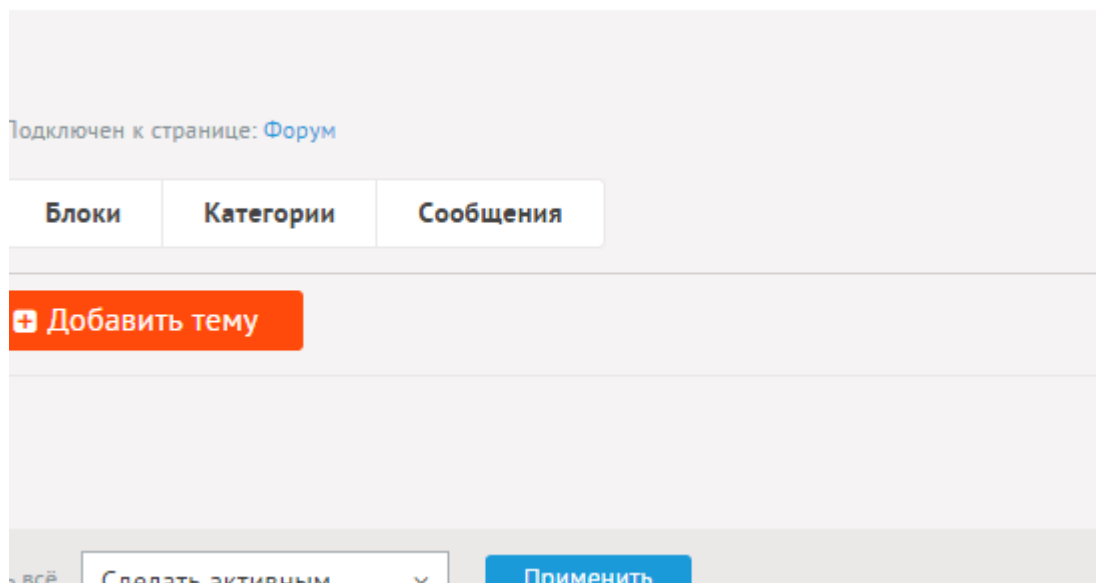
Основные:

- **Название.**
- **Опубликовать на сайте.**
- **Блок форума.**
- **Количество просмотров** – Только для разделов и тем.
- **Псевдоссылка** – ЧПУ (человеко-понятные url url), адрес страницы вида: `http://site.ru/pseudossylka/`. Смотрите параметры сайта.
- **Сортировка: установить перед** – Редактирование порядка следования категории в списке. Поле доступно для редактирования только для категорий, отображаемых на сайте.
- **Время последнего изменения** – Изменяется после сохранения элемента. Отдается в заголовке *Last Modify*.
- **Индексирование для карты сайта** – Категория автоматически индексируется для карты сайта *sitemap.xml*.

Темы



Темы – это страницы, создаваемые пользователями на сайте. Темы объединены общим направлением в категории.



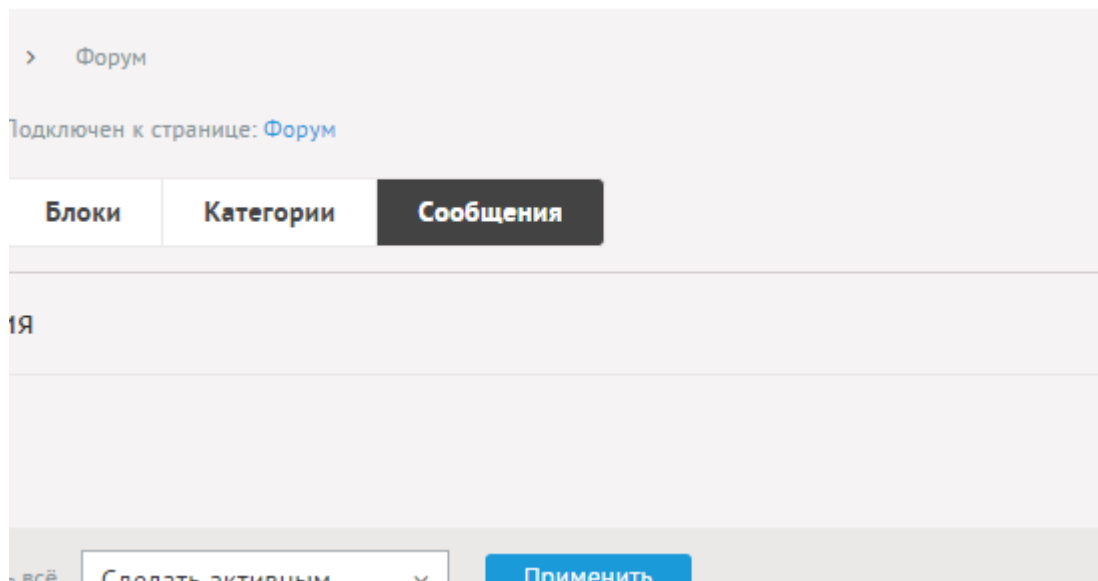
При редактировании для тем можно задать следующие поля.

Основные:

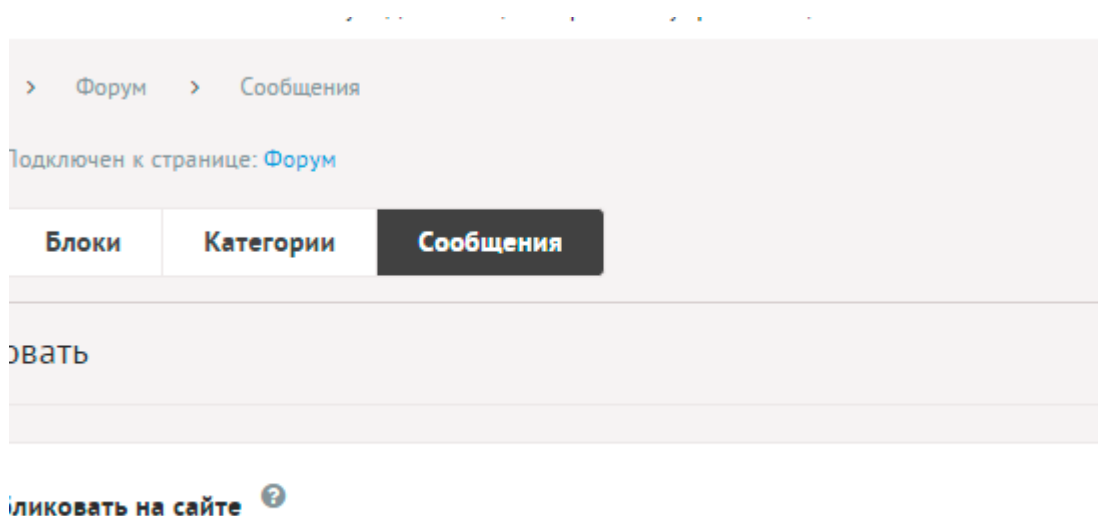
- **Название.**
- **Автор.**
- **Дата** – Вводится в формате дд.мм.гггг чч:мм.
- **Категория.**
- **Опубликовать на сайте.**
- **Закрепить тему (всегда сверху).**
- **Закрывать тему.**
- **Редакция** – Пользователь, отредактировавший тему и дата последней редакции.
- **Количество просмотров.**
- **Псевдоссылка** – ЧПУ (человеко-понятные урл url), адрес страницы вида: *http://site.ru/pseudossylka/*. Смотрите параметры сайта.
- **Индексирование для карты сайта** – Тема автоматически индексируется для карты сайта *sitemap.xml*.

Сообщения

Сообщения создаются пользователями на сайте в одной теме. Сообщения могут быть вложенными, то есть одно сообщение будет ответом для другого. В административной части сообщения выводятся в виде списка в порядке добавления от нового к старому, иерархия (вложенность) при этом не учитывается.



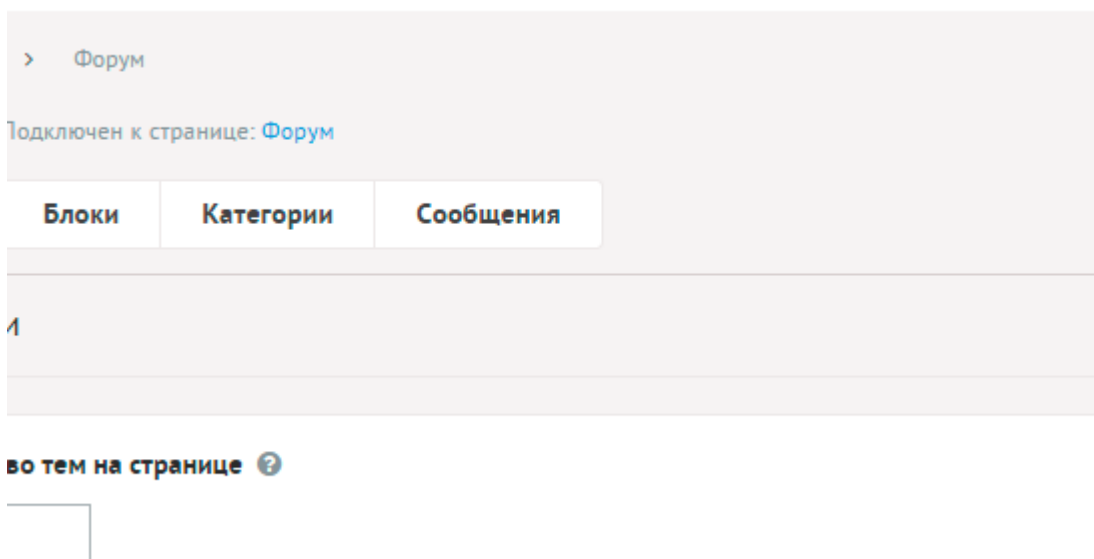
Сообщения имеют следующие характеристики.



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



> Форум

Подключен к странице: Форум

Блоки Категории Сообщения

1

во тем на странице ?

- **Количество тем на странице** – Количество одновременно выводимых тем в списке.
- **Включить «Показать ещё»** – На странице тем появится кнопка «Показать ещё». Увеличивает количество одновременно выводимых тем в списке.
- **Формат даты** – Позволяет настроить отображение даты в модуле. Возможные значения:
 - 01.05.2016,
 - 01.05.2016 14:45,
 - 1 мая 2016 г.,
 - 1 мая,
 - 1 мая 2016, 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 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);
- **block_id** – блоки форума. Идентификаторы блоков перечисляются через запятую. Можно указать отрицательное значение, тогда будут исключены темы из блока. По умолчанию блок не учитывается, выводятся темы из всех блоков;
- **cat_id** – категории форума. Идентификаторы категорий перечисляются через запятую. Можно указать отрицательное значение, тогда будут исключены темы из указанной категории. По умолчанию категория не учитывается, выводятся темы из всех категорий;
- **sort** – сортировка тем: по умолчанию как на странице модуля, **date** – по дате, **rand** – в случайном порядке, **keywords** – темы, похожие по названию для текущей страницы;
- **only_module** – выводить блок только на странице, к которой прикреплен модуль «Форум»: **true** – выводить блок только на странице модуля, по умолчанию блок будет выводиться на всех страницах;
- **defer** – маркер отложенной загрузки шаблонного тега: **event** – загрузка контента только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** – загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка контента только по желанию пользователя;

- **defer_title** – текстовая строка, выводимая на месте появления загружаемого контента с помощью отложенной загрузки шаблонного тега;
- **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

В шаблоне тега можно получить значение любого атрибута через переменную `$result["attributes"]["название атрибута"]`.

show_block_messages – выводит блок сообщений.

Атрибуты:

- **count** – количество выводимых сообщений (по умолчанию 3);
- **block_id** – блоки форума. Идентификаторы блоков перечисляются через запятую. Можно указать отрицательное значение, тогда будут исключены сообщения из указанного блока. По умолчанию блок не учитывается, выводятся сообщения из всех блоков.;
- **cat_id** – категории форума. Идентификаторы категорий перечисляются через запятую. Можно указать отрицательное значение, тогда будут исключены сообщения из указанной категории. По умолчанию категория не учитывается, выводятся сообщения из всех категорий;
- **only_module** – выводить блок только на странице, к которой прикреплен модуль «Форум»: **true** – выводить блок только на странице модуля, по умолчанию блок будет выводиться на всех страницах;
- **defer** – маркер отложенной загрузки шаблонного тега: **event** – загрузка контента только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** – загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка контента только по желанию пользователя;
- **defer_title** – текстовая строка, выводимая на месте появления загружаемого контента с помощью отложенной загрузки шаблонного тега;
- **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">
```

В шаблоне тега можно получить значение любого атрибута через переменную `$result["attributes"]["название атрибута"]`.

База данных

[`{forum}`](#) – Темы форума

[`{forum_blocks}`](#) – Блоки форума

[`{forum_category}`](#) – Категории форума

[`{forum_messages}`](#) – Сообщения на форуме

[`{forum_messages_parents}`](#) – Родительские связи сообщений

[`{forum_show}`](#) – Новые сообщения и темы для пользователей

Файлы

1. `modules/forum/admin/forum.admin.php` – редактирование тем форума;
2. `modules/forum/admin/forum.admin.blocks.php` – редактирование блоков форума;
3. `modules/forum/admin/forum.admin.category.php` – редактирование категорий форума;
4. `modules/forum/admin/forum.admin.config.php` – настройки модуля;
5. `modules/forum/admin/forum.admin.count.php` – количество неактивных тем и сообщений на форуме для меню административной панели;
6. `modules/forum/admin/forum.admin.messages.php` – редактирование сообщений;
7. `modules/forum/admin/forum.admin.role.php` – права пользователя для пользовательской части, используемые модулем;
8. `modules/forum/forum.php` – контроллер;
9. `modules/forum/forum.action.php` – обработка POST-запроса на добавление и редактирование тем;
10. `modules/forum/forum.install.php` – установка модуля;
11. `modules/forum/forum.model.php` – модель;
12. `modules/forum/forum.sitemap.php` – карта ссылок для модуля «Карта сайта»;
13. `modules/forum/img/*` – изображения;
14. `modules/forum/js/forum.js` – JS-сценарий модуля;

15. *modules/forum/views/forum.view.author.php* – шаблон вывода информации о пользователе;
16. *modules/forum/views/forum.view.author_message.php* – шаблон вывода информации о пользователе;
17. *modules/forum/views/forum.view.author_message_main.php* – шаблон вывода информации о пользователе;
18. *modules/forum/views/forum.view.edit.php* – шаблон формы редактирования/добавления категории;
19. *modules/forum/views/forum.view.edit_message.php* – шаблон формы редактирования сообщения;
20. *modules/forum/views/forum.view.first_page.php* – шаблон первой страницы модуля;
21. *modules/forum/views/forum.view.form_message.php* – шаблон формы добавления сообщения;
22. *modules/forum/views/forum.view.form_search.php* – шаблон формы поиска по темам и сообщениям;
23. *modules/forum/views/forum.view.get_attachments.php* – шаблон прикрепленных к сообщению файлов;
24. *modules/forum/views/forum.view.id.php* – шаблон страницы темы;
25. *modules/forum/views/forum.view.id_message.php* – шаблон сообщения;
26. *modules/forum/views/forum.view.id_messages.php* – шаблон сообщения с формой ответа на него и списком ответов;
27. *modules/forum/views/forum.view.list_category.php* – шаблон списка категорий форума;
28. *modules/forum/views/forum.view.list_id.php* – шаблон отдельной темы в категории;
29. *modules/forum/views/forum.view.list_messages.php* – шаблон списка сообщений;
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.rows.php* – шаблон списка категорий форума;
33. *modules/forum/views/forum.view.rows_block.php* – шаблон блока тем форума;
34. *modules/forum/views/forum.view.rows_block_messages.php* – шаблон блока

сообщений форума;

35. *modules/forum/views/forum.view.rows_new.php* – шаблон страницы новых сообщений;

36. *modules/forum/views/forum.view.rows_search.php* – шаблон элементов в списке найденных сообщений;

37. *modules/forum/views/forum.view.show_block.php* – шаблон блока тем форума;

38. *modules/forum/views/forum.view.show_block_messages.php* – шаблон блока сообщений форума.

Опросы

- [Вопросы](#)
- [Варианты пользователей](#)
- [Настройки](#)
- [Шаблонные теги](#)
- [База данных](#)
- [Файлы](#)

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

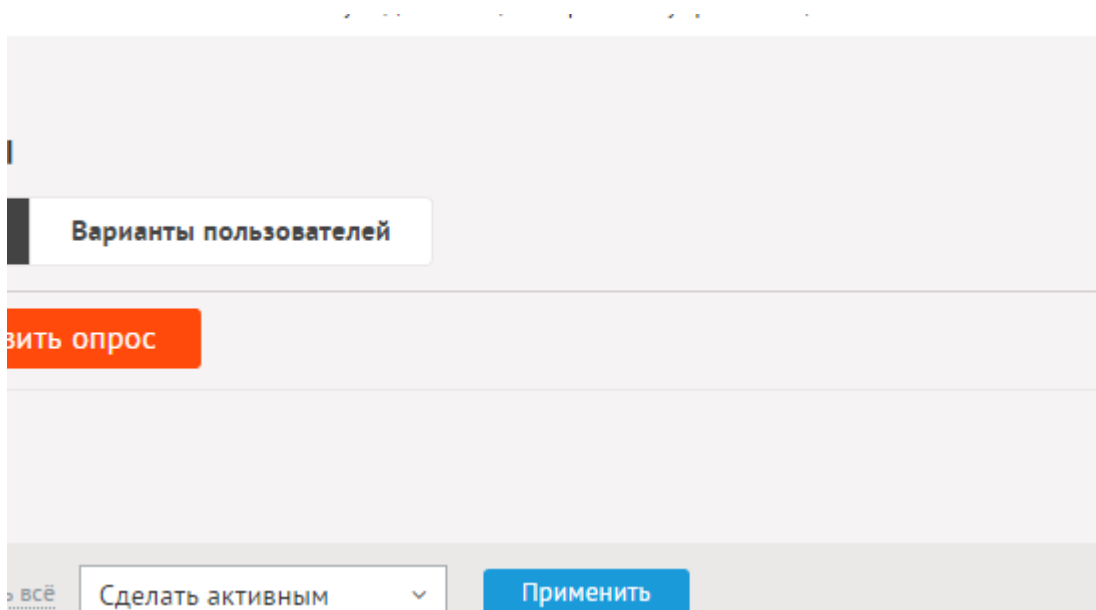
Модуль «Опросы» выводится с помощью шаблонного тега [show_block](#).

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



> Опросы

Варианты пользователей

ОПРАСЫВАТЬ

Какого вы ходите в походы?

Вопросы имеют следующие характеристики.

Варианты пользователей

> Опросы

Варианты пользователей

пользователей

Показать все

Удалить

Применить

Варианты пользователей имеют следующие характеристики.

> [Опросы](#) > [Варианты пользователей](#)

Варианты пользователей

Голосовать

Основные:

- **Ответ** – Ответ пользователя.

Настройки

> [Опросы](#)

Варианты пользователей

1

☐ Только для зарегистрированных пользователей ?

☐ Защита от накруток

- **Только для зарегистрированных пользователей** – Если отмечена, голосовать смогут только зарегистрированные пользователи.
- **Защита от накруток** – Возможные значения:
 - нет,
 - вести лог голосовавших,
 - запрещать голосовать повторно.
- **Использовать защитный код (капчу)** – Для голосования пользователь должен ввести защитный код.
- **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;
- **sort** – сортировка опросов: по умолчанию ручная сортировка как в административной части, **rand** – в случайном порядке;
- **defer** – маркер отложенной загрузки шаблонного тега: **event** – загрузка контента только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** – загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка контента только по желанию пользователя;
- **defer_title** – текстовая строка, выводимая на месте появления загружаемого контента с помощью отложенной загрузки шаблонного тега;
- **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

В шаблоне тега можно получить значение любого атрибута через переменную `$result["attributes"]["название атрибута"]`.

База данных

[{votes}](#) – Опросы

[{votes_answers}](#) – Варианты ответов на опросы

[{votes_site_rel}](#) – Данные о том, на каких страницах сайта выводятся опросы

[{votes_userversion}](#) – Варианты ответов пользователей

Файлы

1. `modules/votes/admin/js/votes.admin.js` – редактирование опросов, JS-сценарий;

2. *modules/votes/admin/votes.admin.php* – редактирование вопросов для голосования;
3. *modules/votes/admin/votes.admin.config.php* – настройки модуля;
4. *modules/votes/admin/votes.admin.userversion.php* – список ответов пользователей;
5. *modules/votes/js/votes.show_block.js* – JS-сценарий модуля;
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.action.php* – обработка запроса при голосовании;
11. *modules/votes/votes.install.php* – установка модуля;
12. *modules/votes/votes.model.php* – модель.

Рассылки

- [Редактирование подписки на сайте](#)
- [E-mail рассылки](#)
- [Категории](#)
- [Подписчики](#)
- [SMS-рассылки](#)
- [Номера телефонов](#)
- [Настройки](#)
- [Шаблонные теги](#)
- [База данных](#)
- [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.

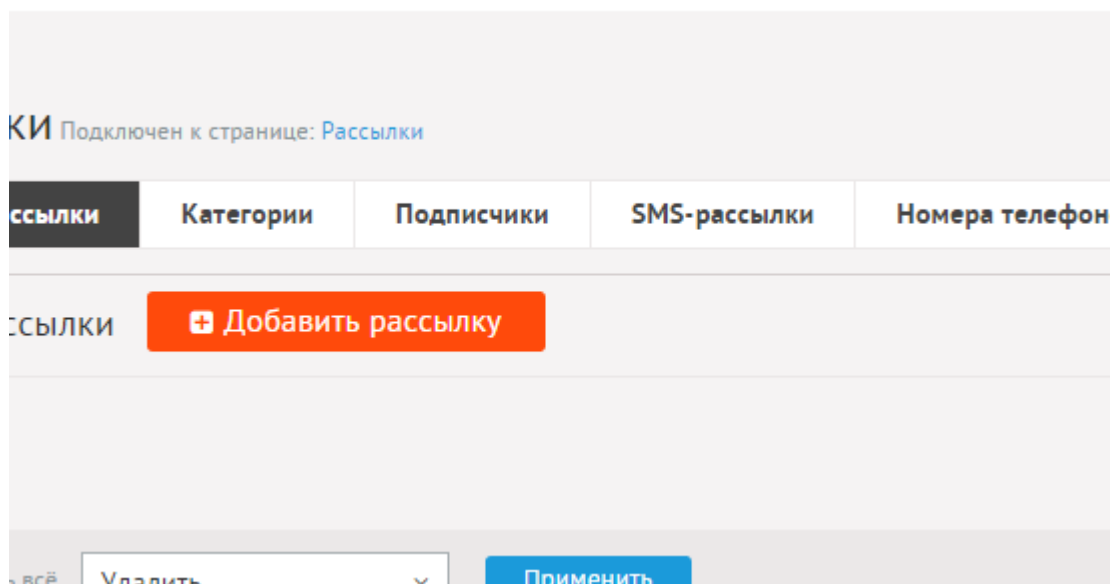
Список подписчиков пополняется следующим образом:

- администратором в разделе «Подписчики»;
- при регистрации пользователя;
- при оформлении заказа;
- электронный адрес добавлен из формы на сайте, которая выводится тегом [show form](#).

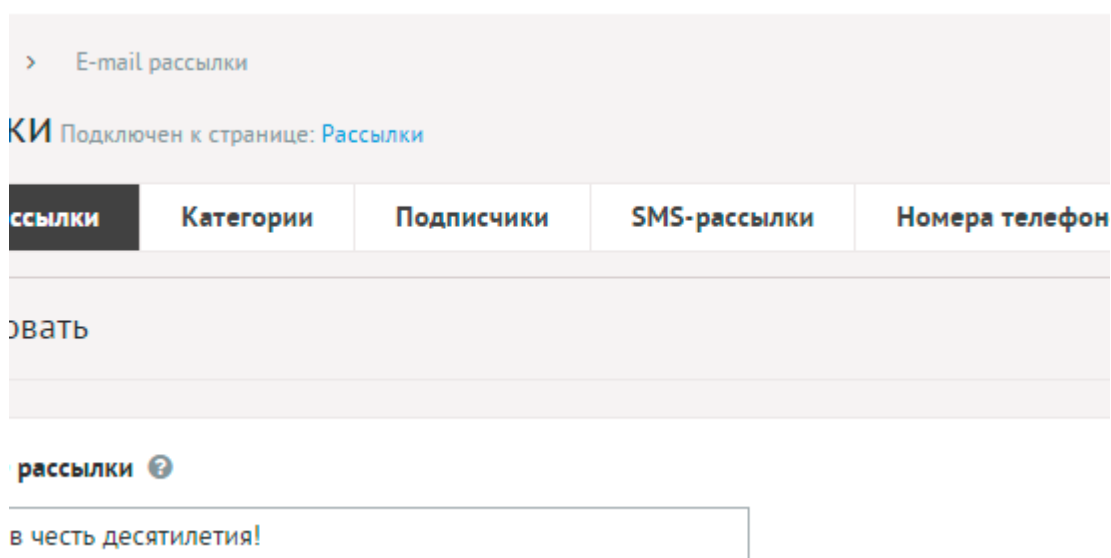
При регистрации и оформлении заказа электронный адрес добавляется в базу автоматически или при отмеченной галке «Подписаться на новости», если это задано в настройках модуля.

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).

E-mail рассылки



Выпуски рассылки выводятся в порядке отправки.



Выпуски имеют следующие характеристики.

Основные:

- **Название рассылки** – Используется в теме письма.
- **Дата добавления** – Отображается дата добавления или отправления рассылки.
- **Отправить рассылку сразу после сохранения** – Если отметить эту галку и сохранить, рассылка начнет отправляться. Если не отмечать галку, рассылка будет сохранена как черновик.
- **Содержимое рассылки** – Текст рассылки. Можно добавить:
 - %name – имя пользователя,
 - %link – ссылка для редактирования категорий рассылки,
 - %actlink – ссылка для отмены рассылки.

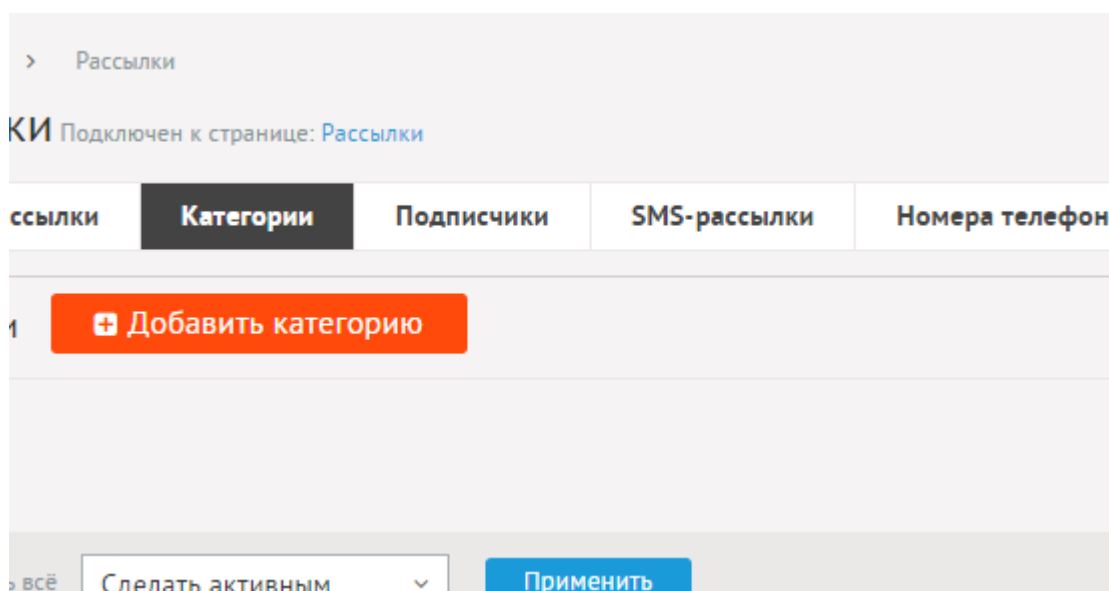
Шаблон и тему письма можно задать в настройках модуля. Если отметить «Применить типограф», контент будет отформатирован согласно правилам экранной типографики с помощью [веб-сервиса «Типограф»](#). Опция «HTML-код» позволяет отключить визуальный редактор для текущего поля. Значение этой настройки будет учитываться и при последующем редактировании.

- **Категория** – Параметр появляется, если в настройках модуля отмечена опция «Использовать категории».

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.

При удалении категории удаляются все вложенные подкатегории и выпуски рассылок.

> Рассылки > Категории

КИ Подключен к странице: [Рассылки](#)

ссылки	Категории	Подписчики	SMS-рассылки	Номера телефон
--------	------------------	------------	--------------	----------------

Добавить

1 компании

Основные:

- **Название** – Параметр имеет разные значения для разных языковых версий сайта.
- **Опубликовать на сайте.**
- **Описание** – Описание категории. Параметр имеет разные значения для разных языковых версий сайта.
- **Сортировка: установить перед** – Редактирование порядка отображения пункта. Поле доступно для редактирования только для незаблокированных категорий.
- **Вложенность: принадлежит** – Категория верхнего уровня.

Подписчики

> Рассылки

КИ Подключен к странице: [Рассылки](#)

ссылки	Категории	Подписчики	SMS-рассылки	Номера телефон
--------	-----------	-------------------	--------------	----------------

КИ [+ Добавить подписчика](#)

всё Применить

Подписчики имеют следующие характеристики.

> Рассылки > Подписчики

КИ Подключен к странице: [Рассылки](#)

ссылки	Категории	Подписчики	SMS-рассылки	Номера телефон
Добавить				

✉ ящик ?

Основные:

- **Почтовый ящик** – Электронный адрес подписчика.
- **Имя получателя** – Можно добавить в рассылку тегом %name.
- **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.

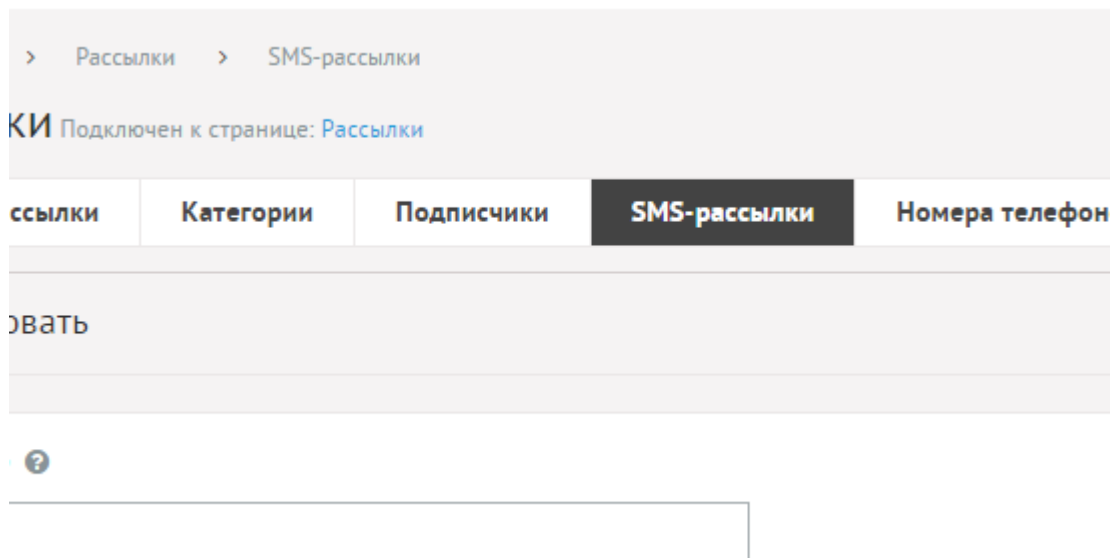
SMS-рассылки

> Рассылки

КИ Подключен к странице: [Рассылки](#)

ссылки	Категории	Подписчики	SMS-рассылки	Номера телефон
рассылки	<div>➕ Добавить SMS-рассылку</div>			

SMS-рассылка возможна, если в [«Параметрах сайта»](#) подключена опция «Подключить SMS-уведомления». Выпуски SMS-рассылки выводятся в порядке отправки.

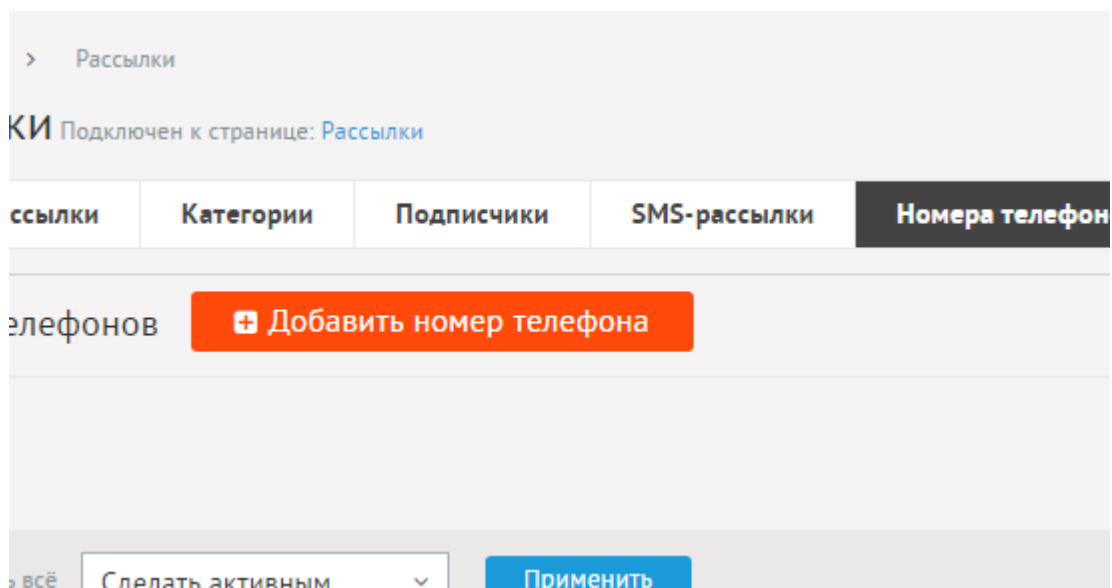


Выпуски SMS-рассылки имеют следующие характеристики.

Основные:

- **Название** – Наименование рассылки. В SMS не подставляется. Используется администратором для удобства.
- **Дата добавления** – Отображается дата добавления или отправления рассылки.
- **Отправить рассылку сразу после сохранения** – 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



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.

> Рассылки > Номера телефонов

КИ Подключен к странице: [Рассылки](#)

Рассылки	Категории	Подписчики	SMS-рассылки	Номера телефонов
Добавить				
Номер телефона в федеральном формате ?				
<input type="text" value="55 4488"/>				

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

> Рассылки

КИ Подключен к странице: [Рассылки](#)

Рассылки	Категории	Подписчики	SMS-рассылки	Номера телефонов
Использовать категории ?				

- **Использовать категории** – Позволяет включить/отключить категории рассылки.
- **Использовать защитный код (капчу)** – Для добавления сообщения пользователь должен ввести защитный код.
- **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.

- **Шаблон письма для рассылки** – Можно добавлять:
 - %title – название сайта,
 - %url – адрес сайта (например, site.ru),
 - %text – текст рассылки,
 - %name – имя пользователя,
 - %link – ссылка для редактирования категорий рассылки,
 - %actlink – ссылка для отмены рассылки.

Параметр имеет разные значения для разных языковых версий сайта.

- **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.
- **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.

- **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.

Шаблонные теги

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

show_form – выводит форму подписки на рассылки.

Атрибуты:

- **defer** – маркер отложенной загрузки шаблонного тега: **event** – загрузка контента только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** – загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка контента только по желанию пользователя;
- **defer_title** – текстовая строка, выводимая на месте появления загружаемого контента с помощью отложенной загрузки шаблонного тега;
- **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">
```

В шаблоне тега можно получить значение любого атрибута через переменную `$result["attributes"]["название атрибута"]`.

База данных

[{subscription}](#) – Рассылки

[{subscription_category}](#) – Категории рассылок

[{subscription_category_parents}](#) – Родительские связи категорий рассылок

[{subscription_category_rel}](#) – Связи рассылок и категорий

[{subscription_emails}](#) – Подписчики на рассылку

[{subscription_emails_cat_unrel}](#) – Отключенные категории рассылок у подписчиков

[{subscription_phones}](#) – Телефоны для рассылок

[{subscription_sms}](#) – SMS-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;

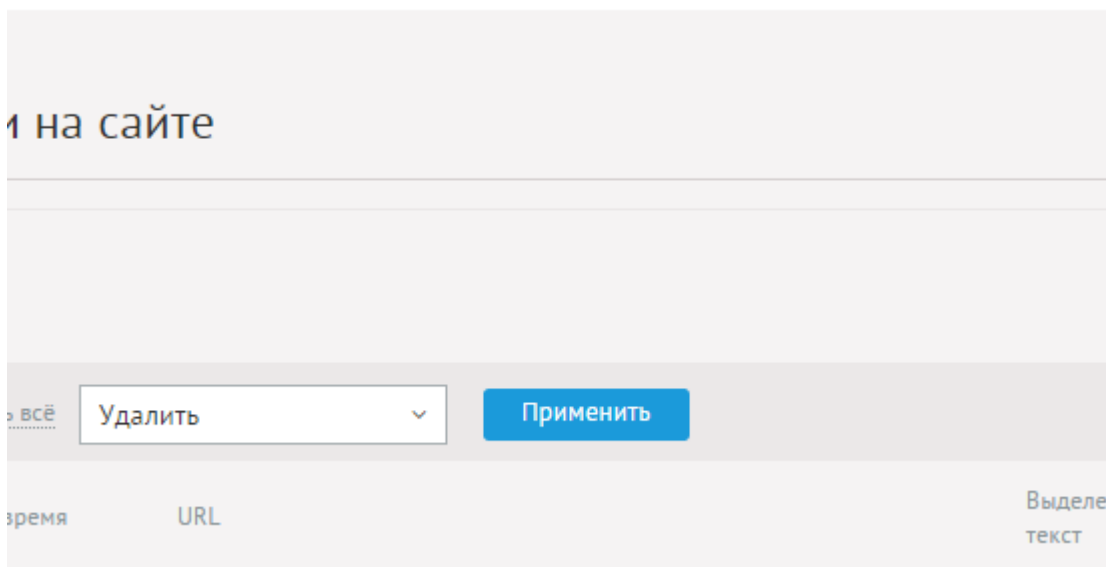
5. *modules/subscription/admin/subscription.admin.phones.php* – база номеров телефонов для SMS рассылки;
6. *modules/subscription/admin/subscription.admin.sms.php* – рассылки по SMS;
7. *modules/subscription/css/subscription.form.css* – {;
8. *modules/subscription/js/subscription.form.js* – JS-сценарий формы подписки на рассылки модуля «Рассылки»;
9. *modules/subscription/subscription.php* – контроллер;
10. *modules/subscription/subscription.action.php* – обработка запроса при отправке сообщения из формы подписки на рассылку;
11. *modules/subscription/subscription.export.php* – экспорт телефонов и электронных ящиков;
12. *modules/subscription/subscription.install.php* – установка модуля;
13. *modules/subscription/subscription.model.php* – модель;
14. *modules/subscription/views/subscription.view.edit.php* – шаблон форма редактирование подписки на новости;
15. *modules/subscription/views/subscription.view.error.php* – шаблон информационного сообщения;
16. *modules/subscription/views/subscription.view.form.php* – шаблон формы подписки на рассылки.

Ошибки на сайте

- [Шаблонные теги](#)
- [База данных](#)
- [Файлы](#)

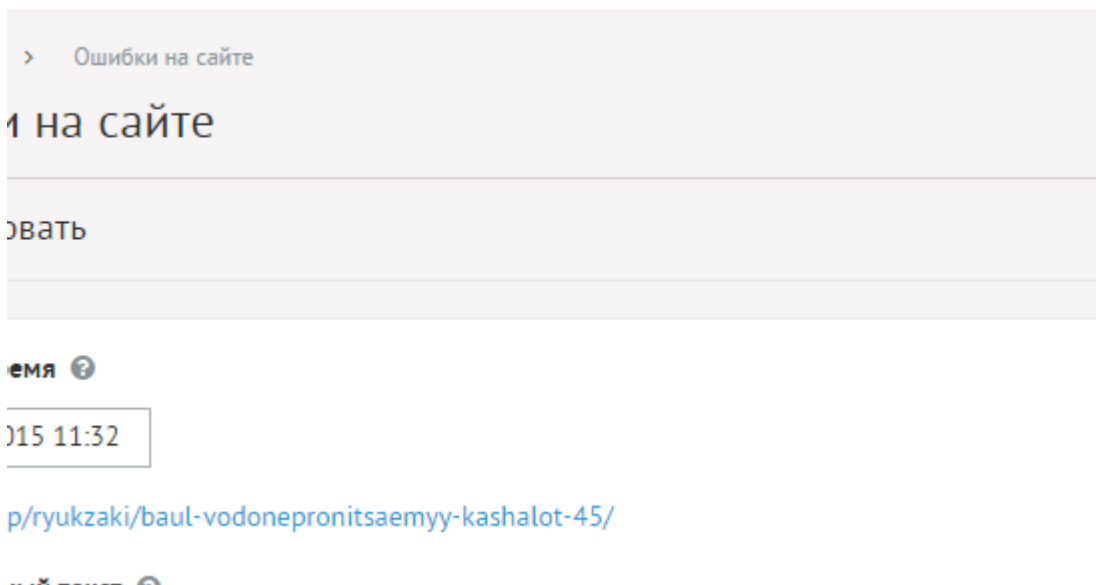
Модуль «Ошибки на сайте» позволяет получать сообщения от пользователей об ошибках в текстах сайта.

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



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.



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».

Атрибуты:

- **defer** – маркер отложенной загрузки шаблонного тега: **event** – загрузка контента только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** – загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка контента только по желанию пользователя;
- **defer_title** – текстовая строка, выводимая на месте появления загружаемого контента с помощью отложенной загрузки шаблонного тега.

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*.

В шаблоне тега можно получить значение любого атрибута через переменную `$result["attributes"][$"название атрибута"]`.

База данных

[{mistakes}](#) – Ошибки на сайте, добавленные пользователями

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/js/mistakes.show_block.js* – JS-сценарий модуля;
4. *modules/mistakes/mistakes.php* – контроллер;
5. *modules/mistakes/mistakes.action.php* – обрабатывает полученные данные из формы;
6. *modules/mistakes/mistakes.install.php* – установка модуля;
7. *modules/mistakes/views/mistakes.view.show_block.php* – шаблон формы добавления уведомления об ошибке.

On-line консультанты

- [Шаблонные теги](#)
- [Файлы](#)

Модуль «On-line консультант» выводит на сайте код одной из предустановленных систем ([JivoSite](#), [LiveTex](#) или [RedHelper](#)) и позволяет установить на сайте форму для связи пользователя с консультантом в режиме on-line.

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

В настройках RedHelper 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 с сайта выбранной системы. Это независимая программа, устанавливаемая на компьютер (Windows+Mac), позволяет наблюдать за посетителями, отвечать на их вопросы и многое другое.

Шаблонные теги

Для работы с модулем «On-line консультант» служат следующие [шаблонные теги](#):

show_block – подключает on-line консультант.

Атрибуты:

- **defer** – маркер отложенной загрузки шаблонного тега: **event** – загрузка контента только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** – загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка контента только по желанию пользователя;
- **defer_title** – текстовая строка, выводимая на месте появления загружаемого

контента с помощью отложенной загрузки шаблонного тега;

— **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" system="jivosite">
```

выведет код для подключения on-line консультанта JivoSite

В шаблоне тега можно получить значение любого атрибута через переменную `$result["attributes"]["название атрибута"]`.

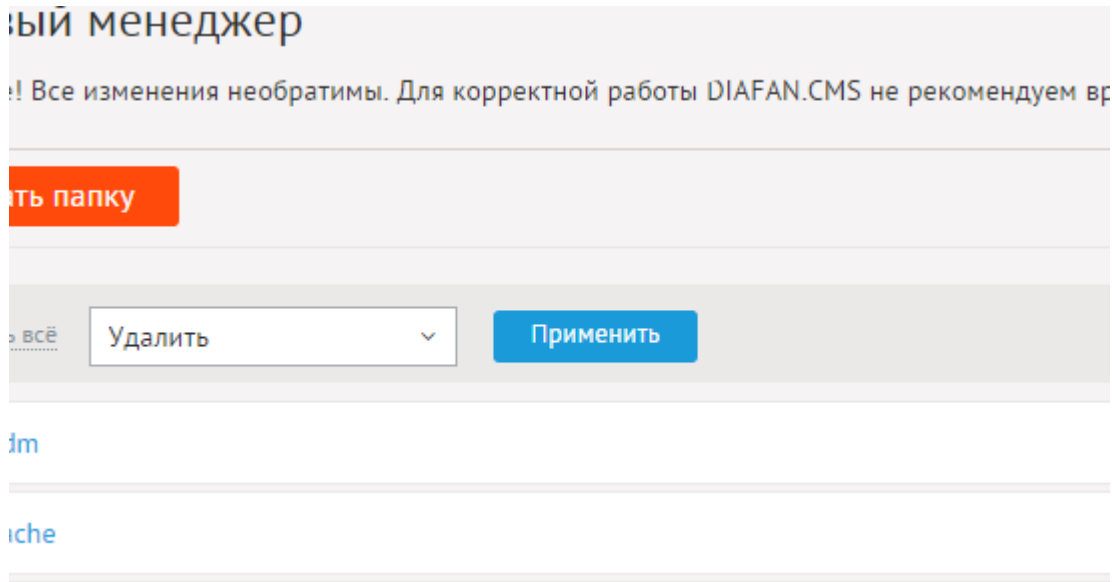
Файлы

1. *modules/consultant/admin/consultant.admin.php* – On-line консультант, система JivoSite;
2. *modules/consultant/consultant.php* – контроллер;
3. *modules/consultant/consultant.install.php* – установка модуля;
4. *modules/consultant/consultant.model.php* – модель;
5. *modules/consultant/views/consultant.view.show_block.php* – шаблон on-line консультанта.

Файловый менеджер

- [Файлы](#)

Модуль «Файловый менеджер» позволяет работать с файловой системой сайта – редактировать скрипты, загружать и удалять файлы.



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

Внимание! Модуль доступен только для веб-мастера, устанавливавшего систему!

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.

Файлы

1. *modules/filemanager/admin/filemanager.admin.php* – файловый менеджер;
2. *modules/filemanager/admin/js/filemanager.admin.js* – файловый менеджер, JS-сценарий;
3. *modules/filemanager/filemanager.install.php* – установка модуля.

Личные сообщения

- [База данных](#)
- [Файлы](#)

Модуль «Личные сообщения» обеспечивает переписку между зарегистрированными пользователями сайта.

Модуль может быть установлен опционально.

Административной части у модуля нет, сообщения видны только пользователям, которые участвуют в переписке.

На сайте модуль подключается к странице сайта.

База данных

[{messages}](#) – Личные сообщения пользователей

[{messages_user}](#) – Контакты пользователей

Файлы

1. *modules/messages/js/messages.js* – JS-сценарий модуля;
2. *modules/messages/messages.php* – контроллер;
3. *modules/messages/messages.action.php* – обработка POST-запроса;
4. *modules/messages/messages.install.php* – установка модуля;
5. *modules/messages/messages.model.php* – модель;
6. *modules/messages/views/messages.view.form.php* – шаблон формы добавления личного сообщения;
7. *modules/messages/views/messages.view.id.php* – шаблон переписки с пользователем;
8. *modules/messages/views/messages.view.list.php* – шаблон списка контактов;
9. *modules/messages/views/messages.view.rows.php* – шаблон элементов в списке контактов;
10. *modules/messages/views/messages.view.rows_id.php* – шаблон переписки с пользователем.

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.

Оплата

Подключение платежных систем и редактирование методов оплаты.

Баланс

Платежные средства на балансе пользователей: пополнение, расходование.

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.

Карта сайта

Ссылки на страницы на отдельной странице сайта и в файле *sitemap.xml*.

Captcha

Формирование защитного кода (капчи) в формах других модулей.

bbCode

Подключение языка разметки bbCode для полей ввода.

Комментарии

- [Подключение](#)
- [Как добавлять комментарии к модулю](#)
- [Комментарии](#)
- [Конструктор формы](#)
- [Настройки](#)
- [Шаблонные теги](#)
- [База данных](#)
- [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 = ''], [string $element_type = 'element'], [integer $site_id = 0], [boolean $ajax = false])` – Показывает комментарии, прикрепленные к элементу, и форму добавления комментария.

- `integer $element_id`: номер элемента, по умолчанию текущий элемент модуля
- `string $module_name`: название модуля, по умолчанию текущий модуль
- `string $element_type`: тип данных
- `integer $site_id`: страница сайта, к которой прикреплен элемент, по умолчанию текущая страница сайта
- `boolean $ajax`: маркер загрузки результатов без перезагрузки страницы

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;`
```

void **element** (array &\$row, array \$params_list, integer \$element_id, string \$module_name, string \$element_type, array \$params_form, [boolean \$hide_form = false]) – Формирует данные об одном комментарии.

- array \$row: массив данных о комментарии
- array \$params_list: дополнительные поля для списка
- integer \$element_id: номер элемента, по умолчанию текущий элемент модуля
- string \$module_name: название модуля, по умолчанию текущий модуль
- string \$element_type: тип данных
- array \$params_form: дополнительные поля для формы
- boolean \$hide_form: скрыть форму ответа на добавленный комментарий

Пример:

```
// получаем данные о комменарии из базы данных
$row = DB::query_fetch_array("SELECT * FROM {comments} WHERE id=5");

// получаем поля, которые выводятся для комментария в списке
$params_list = DB::query_fetch_all("SELECT * FROM {comments_param} WHERE
show_in_list='1'");

// получаем поля, которые используются в форме добавления комментария
$params_form = DB::query_fetch_all("SELECT * FROM {comments_param} WHERE
show_in_form_auth='1' OR show_in_form_no_auth='1'");

// подготавливаем данные для вывода в шаблоне: форматируем данные о
комментарии,
// добавлем форму ответа на комментарий
$this->diafan->_comments->element($row, $params_list, $params_form);
```

array **get_param_values** (integer \$id, array \$params) – Получает дополнительные поля комментариев.

- integer \$id: номер комментария
- array \$params: дополнительные поля комментариев

Пример:

```
// получаем поля, которые выводятся для комментария в списке
$params_list = DB::query_fetch_all("SELECT * FROM {comments_param} WHERE
show_in_list='1'");

// получаем значения дополнительных полей для комментария ID=15
$row["params"] = $this->diafan->_comments->get_param_values(15, $params_list);
```

void **prepare_param_values** (integer \$id) – Запоминает номера комментариев, для которых понадобятся значения доп. полей.

- integer \$id: номер комментария

Пример:

```
// в этом примере будет выполнено три SQL-запроса к базе данных
// для получения значений дополнительных полей ко всем заданным комментариям
$sids = array(3, 5, 7);
foreach($sids as $id)
{
    $params[$id] = $this->diafan->_comments->get_param_values($id, $params_list);
}
```

Пример:

```
// в этом примере будет выполнен один SQL-запрос к базе данных
// для получения значений дополнительных полей ко всем заданным комментариям
$sids = array(3, 5, 7);
foreach($sids as $id)
{
    $this->diafan->_comments->prepare($id);
}
foreach($sids as $id)
{
    $params[$id] = $this->diafan->_comments->get_param_values($id, $params_list);
}
```

void **delete** (integer|array \$element_ids, string \$module_name, [string \$element_type = 'element']) – Удаляет комментарий для одного или нескольких элементов.

- integer|array \$element_ids: номер одного или нескольких элементов
- string \$module_name: название модуля
- string \$element_type: тип данных

Пример:

```
// удалит комментарии к категории новостей ID=3
$this->diafan->_comments->delete(3, 'news', 'cat');

// удалит комментарии к новостям ID=3,4,5
$this->diafan->_comments->delete(array(3, 4, 5) 'news');
```

void **delete_module** (string \$module_name) – Удаляет все комментарии модуля.

- string \$module_name: название модуля

Пример:

```
// удалит комментарии ко всем новостям и категориям новостей
$this->diafan->_comments->delete_module('news');
```

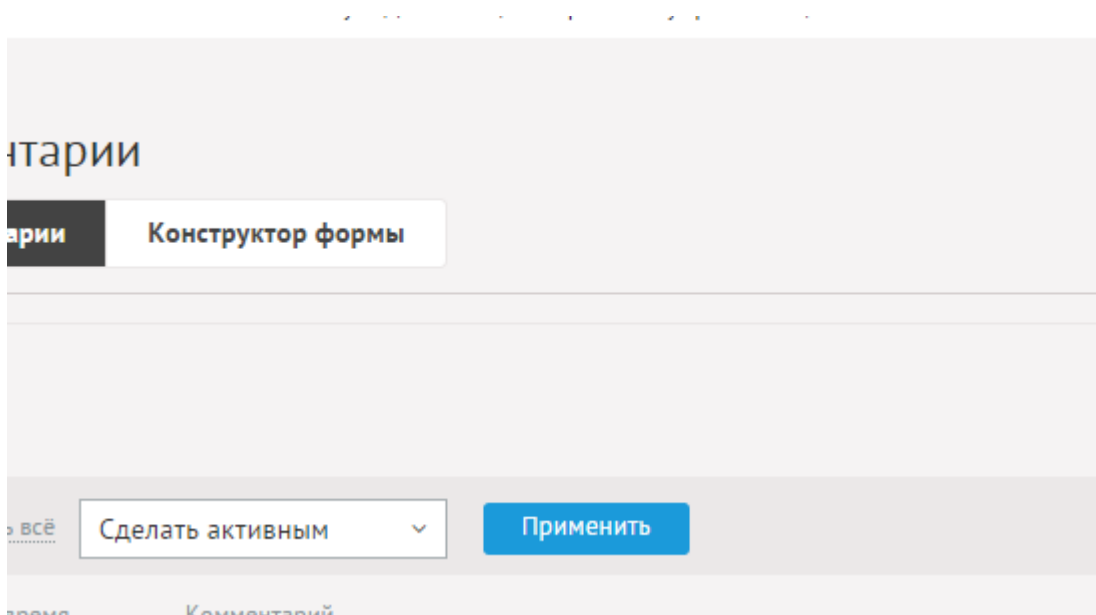

Как добавлять комментарии к модулю

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:

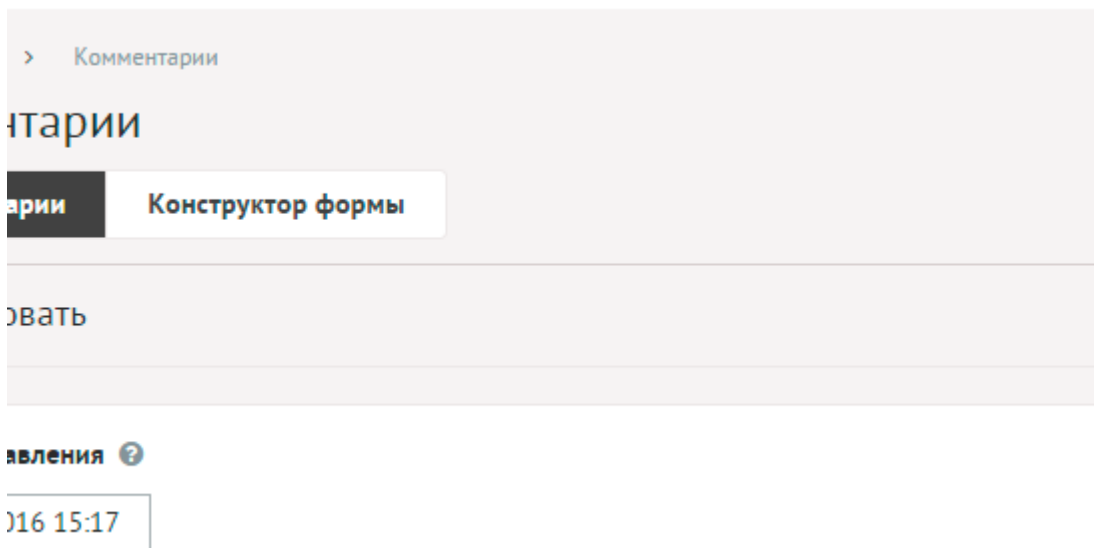
```
public $variables = array(  
    'main' => array(  
        'comments' => 'module',  
        ...  
    ),  
    ...  
);
```

Комментарии



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.

При удалении какого-либо элемента, удаляются все комментарии, прикрепленные к нему.

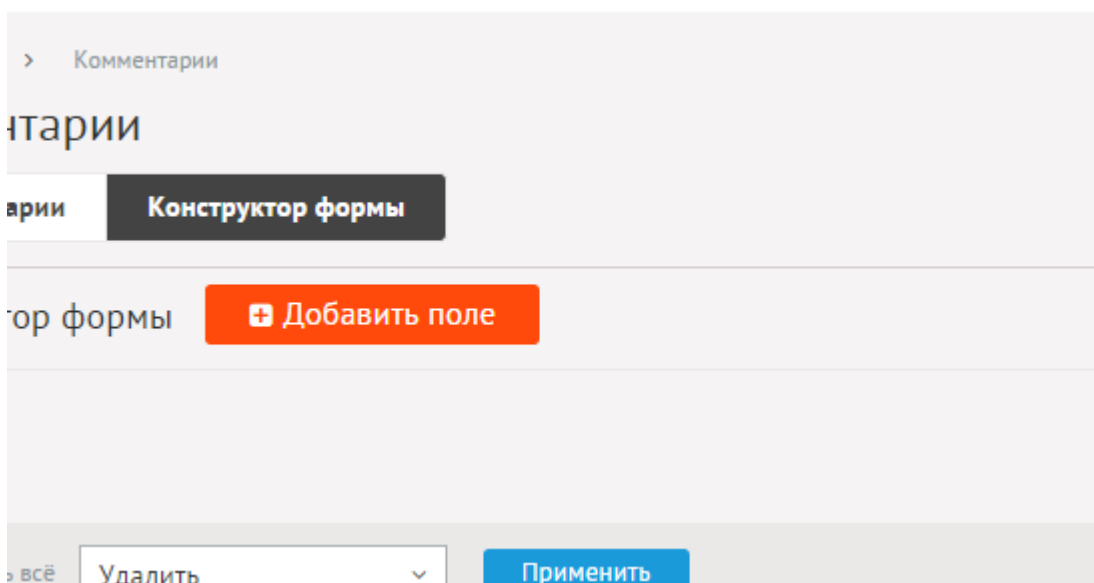


Комментарии имеют следующие характеристики.

Основные:

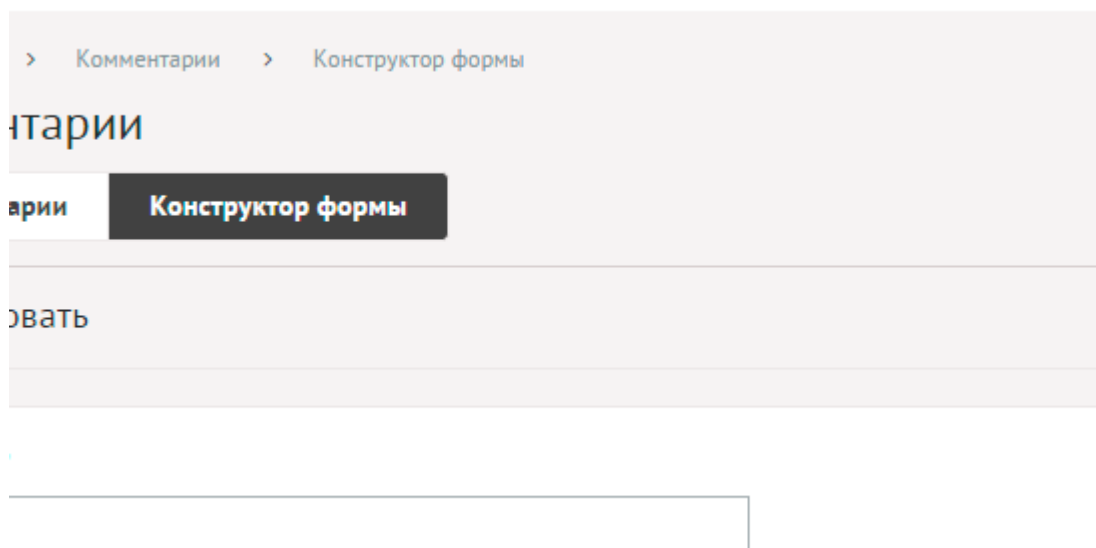
- **Дата добавления** – В формате дд.мм.гггг чч:мм.
- **Пользователь** – Пользователь, добавивший комментарий (если комментарий добавлен зарегистрированным пользователем).
- **Комментарий.**
- **Дополнительные поля** – Поля, добавленные в конструкторе формы.
- **Опубликовать на сайте** – Если не отмечена, 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



Количество и содержание полей комментариев может быть различно. Это обеспечивает конструктор формы.

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,
 - поле с редактором,
 - галочка,
 - переключатель,
 - выпадающий список,
 - список с выбором нескольких значений,
 - электронный ящик,
 - телефон,
 - 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.

Настройки

> Комментарии

Комментарии

Комментарии Конструктор формы

1

Даты ?

5:30

- **Формат даты** – Позволяет настроить отображение даты в модуле. Возможные значения:
 - 01.05.2016,

- 01.05.2016 14:45,
- 1 мая 2016 г.,
- 1 мая,
- 1 мая 2016, понедельник,
- вчера 15:30,
- не отображать.
- **Максимальная вложенность** – Ограничивает вложенность дерева комментариев.
- **Количество комментариев на странице** – Количество комментариев первого уровня, показываемых на одной странице.
- **Включить «Показать ещё»** – На странице комментариев появится кнопка «Показать ещё». Увеличивает количество одновременно выводимых комментариев в списке.
- **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:
%message

 Отписаться можете по ссылке.

- **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:

Здравствуйтесь, администратор сайта %title (%url)!
На странице %urlpage появился новый комментарий:
%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;
- **element_type** – тип данных (*element* – элемент (по умолчанию), *cat* – категория);
- **sort** – сортировка комментариев: **date** – по дате (по умолчанию), **rand** – в случайном порядке;
- **defer** – маркер отложенной загрузки шаблонного тега: **event** – загрузка контента только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** – загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка контента только по желанию пользователя;
- **defer_title** – текстовая строка, выводимая на месте появления загружаемого контента с помощью отложенной загрузки шаблонного тега;
- **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

В шаблоне тега можно получить значение любого атрибута через переменную `$result["attributes"]["название атрибута"]`.

База данных

[{comments}](#) – Комментарии

[{comments_mail}](#) – Подписка пользователей на новые комментарии

[{comments_param}](#) – Дополнительные поля комментариев

[{comments_param_element}](#) – Значения дополнительных полей комментариев

[{comments_param_select}](#) – Варианты значения дополнительных полей комментариев с типом список

[{comments_parents}](#) – Родительские связи комментариев

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.dashboard.php* – комментарии для событий;
5. *modules/comments/admin/comments.admin.inc.php* – подключение модуля к административной части других модулей;
6. *modules/comments/admin/comments.admin.param.php* – конструктор комментариев;
7. *modules/comments/admin/js/comments.admin.inc.config.js* – поле "Комментарии", JS-сценарий;
8. *modules/comments/admin/js/comments.admin.param.js* – конструктор комментариев, JS-сценарий;
9. *modules/comments/comments.php* – контроллер;
10. *modules/comments/comments.action.php* – обработка запроса при добавления комментария;
11. *modules/comments/comments.inc.php* – подключение модуля;

12. *modules/comments/comments.install.php* – установка модуля;
13. *modules/comments/comments.model.php* – модель;
14. *modules/comments/comments.rss.php* – RSS лента комментариев;
15. *modules/comments/js/comments.get.js* – JS-сценарий модуля;
16. *modules/comments/views/comments.view.form.php* – шаблон формы добавления комментария;
17. *modules/comments/views/comments.view.get.php* – шаблон вывода комментариев;
18. *modules/comments/views/comments.view.id.php* – шаблон одного комментария;
19. *modules/comments/views/comments.view.images.php* – шаблон прикрепленных к комментарию изображения, выводимых в форме добавления комментария;
20. *modules/comments/views/comments.view.list.php* – шаблон списка комментариев;
21. *modules/comments/views/comments.view.rows_block.php* – шаблон блока комментариев;
22. *modules/comments/views/comments.view.show_block.php* – шаблон блока комментариев;
23. *modules/comments/views/comments.view.show_block_main.php* – шаблон блока комментариев.

Рейтинг

- [Подключение](#)
- [Как добавлять рейтинг к модулю](#)
- [Административная часть](#)
- [Настройки](#)
- [База данных](#)
- [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 = "], [string \$element_type = 'element'], [integer \$site_id = 0], [boolean \$full = false]) – Показывает рейтинг для элемента.

- integer \$element_id: номер элемента модуля, по умолчанию текущий элемент модуля
- string \$module_name: название модуля, по умолчанию текущий модуль
- string \$element_type: тип данных
- integer \$site_id: страница сайта, к которой прикреплен элемент, по умолчанию текущая страница сайта
- boolean \$full: полная информация

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;
```

void **prepare** ([integer \$element_id = 0], [string \$module_name = "], [string \$element_type = 'element']) – Запоминает данные элемента, которому нужно будет вывести рейтинг.

- integer \$element_id: номер элемента модуля, по умолчанию текущий элемент модуля

- `strint $module_name`: название модуля, по умолчанию текущий модуль
- `string $element_type`: тип данных

Пример:

```
// в этом примере будет выполнено три SQL-запроса к базе данных
// для получения рейтинга всех заданных товаров
$ids = array(3, 5, 7);
foreach($ids as $id)
{
    $rating[$id] = $this->diafan->_rating->get($id, 'shop');
}
```

Пример:

```
// в этом примере будет выполнен один SQL-запрос к базе данных
// для получения рейтинга всех заданных товаров
$ids = array(3, 5, 7);
foreach($ids as $id)
{
    $this->diafan->_rating->prepare($id, 'shop');
}
foreach($ids as $id)
{
    $rating[$id] = $this->diafan->_rating->get($id, 'shop');
}
```

void delete (integer|array \$element_ids, strint \$module_name, [string \$element_type = 'element']) – Удаляет рейтинг для одного или нескольких элементов.

- `integer|array $element_ids`: номер одного или нескольких элементов
- `strint $module_name`: название модуля
- `string $element_type`: тип данных

Пример:

```
// удалит рейтинг категории новостей ID=3
$this->diafan->_rating->delete(3, 'news', 'cat');

// удалит рейтинг к новостям ID=3,4,5
$this->diafan->_rating->delete(array(3, 4, 5) 'news');
```

void delete_module (string \$module_name) – Удаляет все рейтинги элементов модуля.

- `string $module_name`: название модуля

Пример:

```
// удалит рейтинг всех новостей и категорий новостей
$this->diafan->_rating->delete_module('news');
```

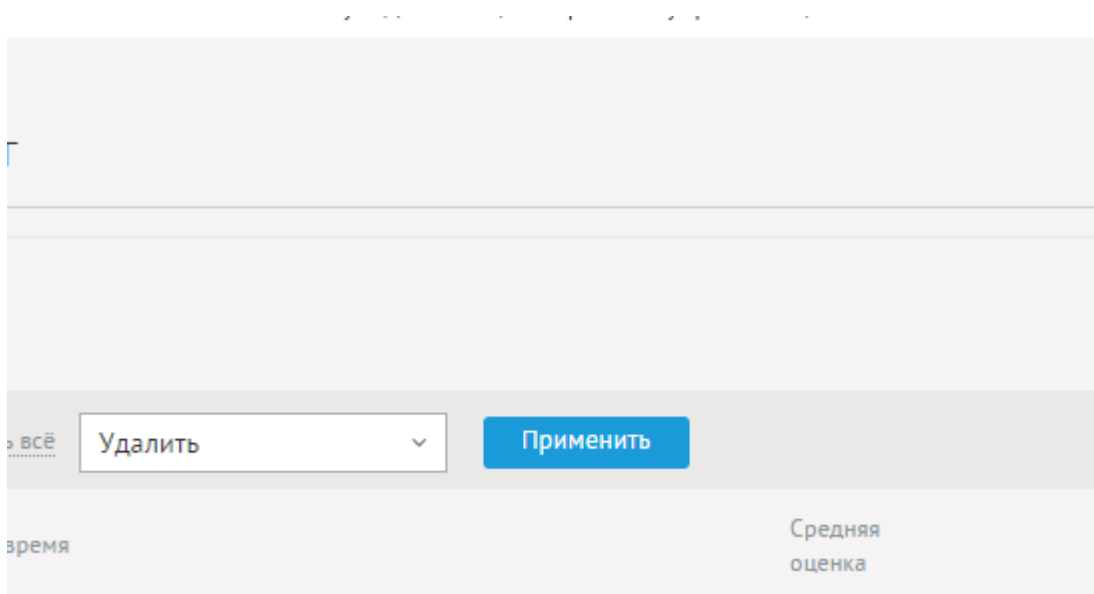
Как добавлять рейтинг к модулю

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



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.

> Рейтинг

Название	Модуль

Оценка ?

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

> Рейтинг

Название	Модуль

ко для зарегистрированных пользователей ?

г накруток

ать голосовать повторно

- **Только для зарегистрированных пользователей** – Параметр позволяет запретить неавторизованным пользователям голосовать.

- **Защита от накруток** – Возможные значения:
 - нет,
 - вести лог голосовавших,
 - запрещать голосовать повторно.

База данных

[{rating}](#) – Рейтинг элементов модулей

Файлы

1. *modules/rating/admin/js/rating.admin.inc.config.js* – поле "Рейтинг", JS-сценарий;
2. *modules/rating/admin/rating.admin.php* – редактирование рейтингов;
3. *modules/rating/admin/rating.admin.config.php* – настройки модуля;
4. *modules/rating/admin/rating.admin.inc.php* – подключение модуля к административной части других модулей;
5. *modules/rating/img/** – изображения;
6. *modules/rating/js/rating.get.js* – JS-сценарий модуля;
7. *modules/rating/rating.php* – контроллер;
8. *modules/rating/rating.action.php* – обрабатывает полученные данные из формы;
9. *modules/rating/rating.inc.php* – подключение модуля;
10. *modules/rating/rating.install.php* – установка модуля;
11. *modules/rating/views/rating.view.get.php* – шаблон рейтинга элемента.

Перелинковка

- [Подключение](#)
- [Как добавлять перелинковку к модулю](#)
- [Ключевые слова](#)
- [Импорт/экспорт](#)
- [Настройки](#)
- [База данных](#)
- [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

`void get (string &$text)` – Подставляет ключевые слова.

- string \$text: исходный текст

Пример:

```
// заменяем в тексте ключевые слова на ссылки
$this->diafan->_keywords->get($text);
```

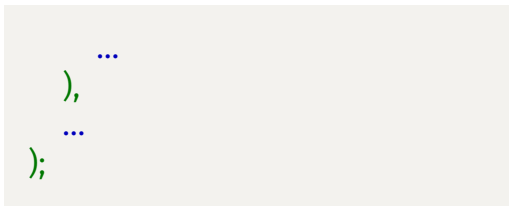
Если в настройках модуля подключен модуль «Перелинковка», то ссылки подставляются в тексты, обрабатываемые функцией [htmleditor](#).

Как добавлять перелинковку к модулю

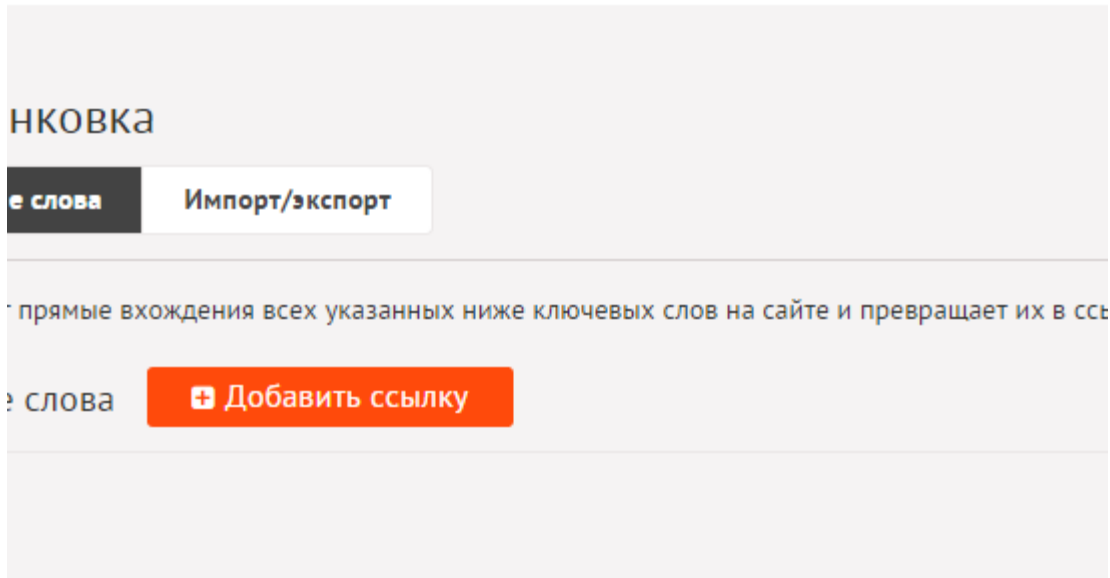
В настройках модуля (*modules/модуль/admin/модуль.admin.config.php*) необходимо добавить параметр **'keywords'**:

Пример:

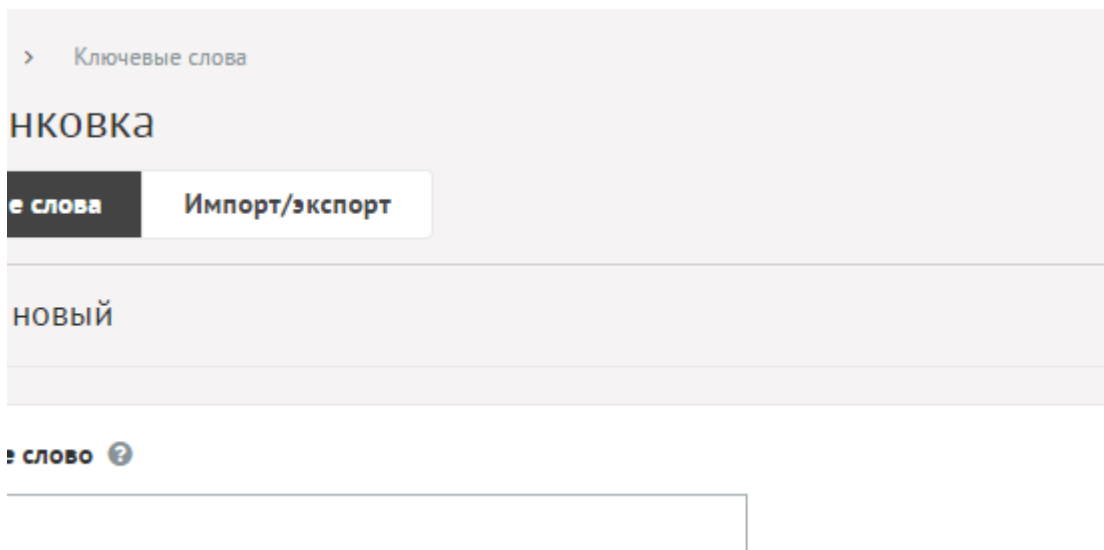
```
public $variables = array(
    'main' => array(
        'keywords' => 'module',
```



Ключевые слова



Ключевые слова и ссылки к ним можно редактировать непосредственно из списка.



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

> Перелинковка

НКОВКА

е слова **Импорт/экспорт**

ить неописанные в файле строки

ор Файл не выбран

ировать

Экспорт

Скачать файл

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

> Перелинковка

ПЕРЛИНГОВКА

Ключевые слова Импорт/экспорт

1

Максимальное количество ссылок на странице ?

В настройках можно указать:

- **Максимальное количество ссылок на странице** – Ограничивает количество ссылок на одной странице.

База данных

[{keywords}](#) – Ключевые слова для перелинковки

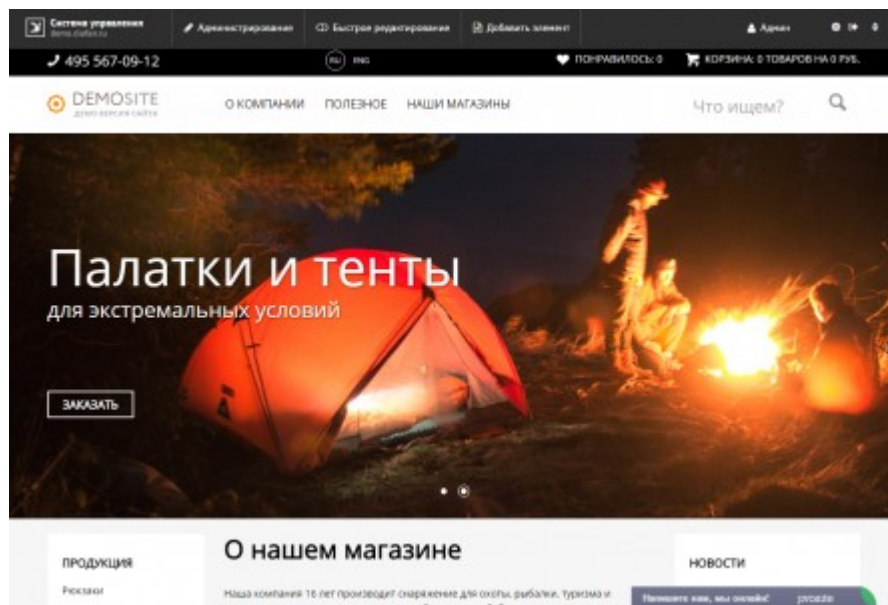
Файлы

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.



Внимание! Для обеспечения редактирования элементов на страницах сайта модуль оборачивает все редактируемые HTML-элементы в дополнительные слои вида ``. В зависимости от верстки и CSS-оформления пользовательских шаблонов дизайна, внешний вид сайта с подключенной панелью быстрого редактирования (каким его видит только авторизованный администратор) может несколько отличаться от внешнего вида без панели, в каком сайт видят все пользователи.

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 $table_name, [integer $lang_id = 0], [string $type = ""])` – Генерирует ссылку на форму редактирования.

- `string $text`: значение переменной
- `string $name`: название переменной
- `integer $element_id`: номер элемента
- `string $table_name`: таблица

- 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.

- string \$value: текущий перевод
- string \$name: строка для перевода
- string \$module_name: модуль

Пример:

```
$text = $this->diafan->_useradmin->get_lang('hello', 'привет', 'site');
```

array|boolean false **get_meta (integer \$element_id, string \$module_name)** – Получает ссылки для редактирования мета-данных через панель администрирования.

- integer \$element_id: номер элемента
- string \$module_name: модуль

Пример:

```
// получаем ссылки для редактирования мета-данных для страницы новости ID=3
$useradmin_links = $this->diafan->_useradmin->get_meta(3, "news");

echo '<meta name="useradmin_title" content="'. $useradmin_links["title_meta"]. "'>';
echo '<meta name="useradmin_description" content="'. $useradmin_links["descr"]. "'>';
echo '<meta name="useradmin_keywords" content="'. $useradmin_links["keywords"]. "'>';
```

void **edit ()** – Генерирует данные для формы редактирования.

Пример:

```
echo $this->diafan->_useradmin->edit();
// выведет форму редактирования данных, заданных в $_GET массиве
```

string **type (string \$name)** – Возвращает тип данных по имени переменной.

- string \$name: имя редактируемой переменной

Пример:

```
echo 'Тип данных поля created: ' . $this->diafan->_useradmin->type("created");
// выведет:
// Тип данных поля created: date
```

Для разработчика

Используется в основном в [моделях](#) модулей при генерации данных, реже в [шаблоне](#) для предоставления доступа к панели быстрого редактирования.

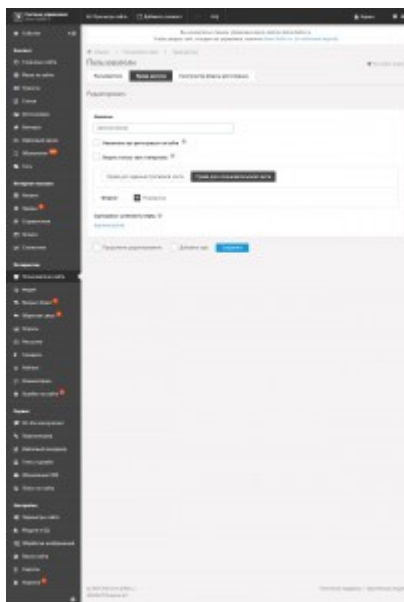
Работа с модулем должна производиться вне зоны кэширования.

Пример:

```
// news.model.php
if(! $this->result = $this->diafan->_cache->get($cache_meta, 'news'))
{
    $this->diafan->_cache->save($this->result, $cache_meta, 'news');
}
//...
foreach ($this->result["rows"] as &$row)
{
    //...
    $row["name"] = $this->diafan->_useradmin->get(
        $row["name"], 'name', $row["id"], 'news', _LANG
    );
    $row["anons"] = $this->diafan->_useradmin->get(
        $row["anons"], 'anons', $row["id"], 'news', _LANG
    );
    $row["date"] = $this->diafan->_useradmin->get(
        $row["date"], 'created', $row["id"], 'news'
    );
    //...
}
```

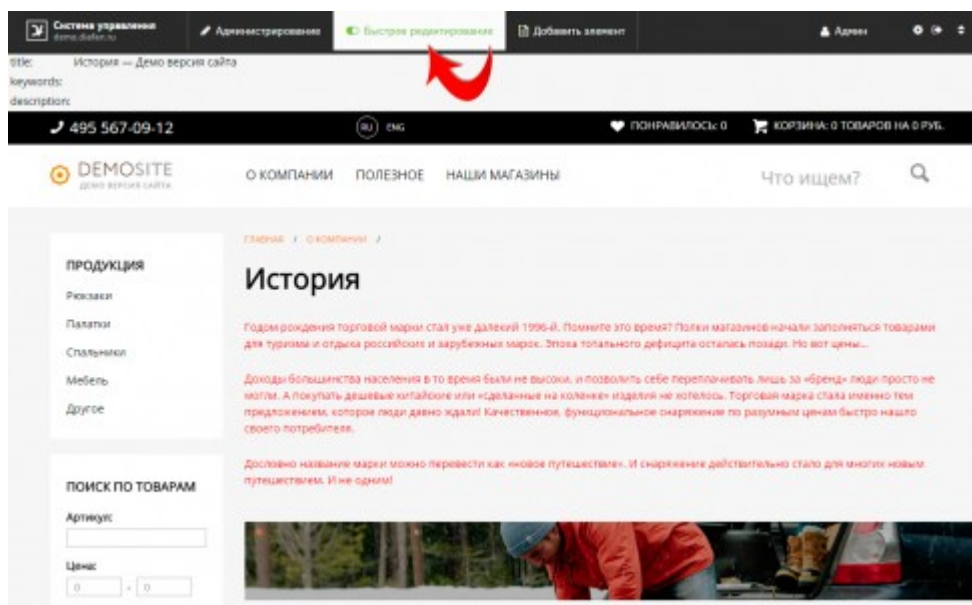
Как работает панель быстрого редактирования

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.

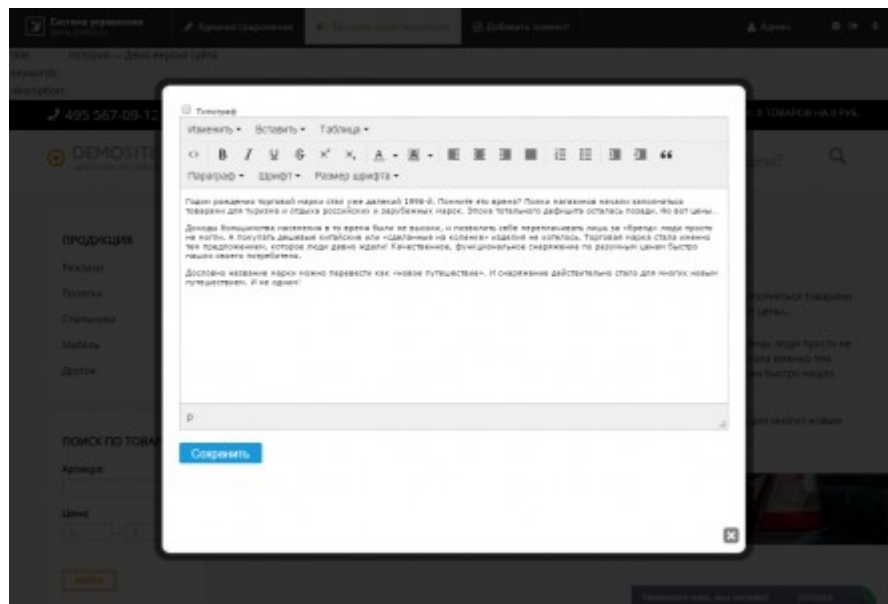


Внимание! Веб-мастер должен установить скрипты для работы панели редактирования см. [Подключение](#).

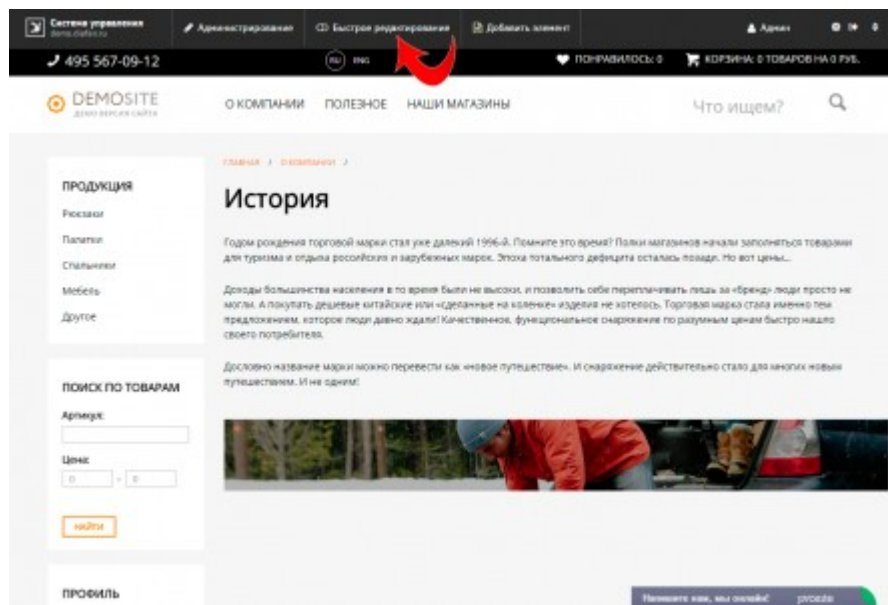
После включения режима редактирования в административной части, на сайте сверху появится административная панель. На этой панели нужно включить режим редактирования. Режим включен, когда подсвечен красным цветом.



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.action.php` – обработка запроса на редактирование данных из пользовательской части;
2. `modules/useradmin/js/useradmin.edit.js` – форма редактирования контента из пользовательской части, JS-сценарий;
3. `modules/useradmin/js/useradmin.js` – панель быстрого редактирования, JS-сценарий;
4. `modules/useradmin/useradmin.php` – контроллер;

5. *modules/useradmin/useradmin.action.php* – подгрузка панели быстрого редактирования;
6. *modules/useradmin/useradmin.css* – панель быстрого редактирования, CSS;
7. *modules/useradmin/useradmin.edit.php* – форма редактирования контента из пользовательской части;
8. *modules/useradmin/useradmin.edit.css* – форма редактирования контента из пользовательской части, CSS;
9. *modules/useradmin/useradmin.inc.php* – подключение модуля;
10. *modules/useradmin/useradmin.install.php* – установка модуля;
11. *modules/useradmin/views/m/useradmin.view.panel.php* – шаблон панели быстрого редактирования;
12. *modules/useradmin/views/useradmin.view.edit.php* – шаблон формы редактирования данных;
13. *modules/useradmin/views/useradmin.view.get.php* – шаблон данных, доступных для редактирования с помощью панели быстрого редактирования;
14. *modules/useradmin/views/useradmin.view.panel.php* – шаблон панели быстрого редактирования.

Оплата

- [Подключение](#)
- [Структура бэкенда](#)
- [Модуль Webmoney](#)
- [Модуль Баланс пользователя](#)
- [Модуль Банковские платежи](#)
- [Модуль QIWI](#)
- [Модуль Яндекс.Касса](#)
- [Платежные агрегаторы](#)
- [Методы оплаты](#)
- [История платежей](#)
- [База данных](#)
- [Файлы](#)

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

Модуль описывает работу с платежными системами. Используется в других модулях в виде подключения. В стандартной комплектации DIAFAN.CMS «Оплата» подключена к корзине интернет-магазина и балансу пользователя.

Подключение

Подключаемая часть – файл *modules/payment/payment.inc.php*. В нем описан класс *Payment_inc*. В модуле к объекту класса можно обратиться через переменную `$this->diafan->_payment`. Экземпляр класса создается при первом вызове переменной.

Методы

array **get_all** ([string \$where = ""]) – Получает список платежных систем.

- string \$where: условие для SQL-запроса

Пример:

```
$rows = $this->diafan->_payment->get_all();
print_r($rows);
// выведет:
// (
//   [0] => Array
//   (
//       [id] => 1
//       [name] => Наличными курьеру
//       [text] => Заказ необходимо оплатить курьеру на руки наличными
//       [payment] =>
//   )
//   [1] => Array
//   (
```

```
//      [id] => 3
//      [name] => WebMoney
//      [text] => Используйте мгновенную оплату заказа через систему WebMoney. Это
ускорит доставку Вашего заказа.
//      [payment] => webmoney
//  )
// ...
//)
```

array **get (integer \$id)** – Возвращает название метода оплаты и его тип по ID.

- integer \$id: ид метода оплаты

Пример:

```
$row = $this->diafan->_payment->get(3);
print_r($rows);
// выведет:
//(
// [name] => WebMoney
// [payment] => webmoney
//)
```

integer **add_pay (integer \$element_id, string \$module_name, integer \$payment_id, float \$summ)** – Добавление записи о новом платеже в историю платежей, возвращает ID записи.

- integer \$element_id: номер элемента, для которого будет совершен платеж
- string \$module_name: модуль, осуществляющий запрос
- integer \$payment_id: ID способа оплаты
- float \$summ: сумма платежа

Пример:

```
// добавляет платеж на заказ в интернет магазине №12 на сумму 300 руб. через
платежную систему WebMoney (ID=3, из предыдущего примера)
$pay_id = $this->diafan->_payment->add_pay(12, 'cart', 3, 300);
echo 'Номер платежа: '.$pay_id;
// выведет:
// Номер платежа: 13
```

integer **update_pay (integer \$element_id, string \$module_name, integer \$payment_id, float \$summ)** – Добавление или обновление записи о платеже в историю платежей, возвращает ID записи.

- integer \$element_id: номер элемента, для которого будет совершен платеж
- string \$module_name: модуль, осуществляющий запрос
- integer \$payment_id: ID способа оплаты
- float \$summ: сумма платежа

Пример:

```
// изменяем сумму платежа из предыдущего примера на сумму 400 руб.
$pay_id = $this->diafan->_payment->update_pay(12, 'cart', 3, 400);
echo 'Номер платежа, данные которого изменены: '.$pay_id;
// выведет:
// Номер платежа, данные которого изменены: 13
```

array **get_pay** (string \$element_id, string \$module_name, [string \$code = '']) – Генерирует данные для второго шага в оформлении заказа: оплата.

- string \$element_id: ID заказа
- string \$module_name: модуль, осуществляющий запрос
- string \$code: код доступа

Пример:

```
// получит данные для оплаты заказа №6
$result = $this->diafan->_payment->get_pay(6, 'cart');
```

array **check_pay** (integer \$id, string \$payment) – Проверяет наличие платежа, используется в конкретном методе оплаты.

- integer \$id: номер платежа
- string \$payment: платежная система

Пример:

```
// если платежа нет в базе данных, то выполнение скрипта остановиться и
// пользователь увидит ошибку 404
$pay = $this->diafan->_payment->check_pay(12, 'webmoney');
```

void **success** (array \$pay, [string \$type = 'all']) – Действия при успешной оплате.

- array \$pay: данные платежа
- string \$type: тип операции: all - все действия, pay - оплата, redirect - редирект на страницу платежа

Пример:

```
// если платеж успешно проведен, то
if(проверка успешного проведения платежа)
{
    $this->diafan->_payment->success($pay);
}
```

void **fail** (array \$pay) – Действия при неудачной оплате.

- array \$pay: данные платежа

Пример:

```
// если платеж успешно проведен, то
if(! проверка успешного проведения платежа)
{
    $this->diafan->_payment->fail($pay);
}
```

Структура бэкенда

К модулю можно подключить неограниченное количество бэкендов – модулей для работы с конкретными платежными системами.

Для этого нужно разработать несколько файлов стандартной структуры, учитывающие особенности платежной системы и положить папку с этими файлами в папку *payment/backend*. Далее создать метод оплаты, в котором в качестве платежной системы указать созданный бэкенд.

Платежный модуль (или бэкенд) должен иметь уникальное название латинскими буквами (по примеру *robokassa*, *webmoney*, *balance* и др.). Для примера будем использовать название *paymethod*.

В папке *payment/backend/paymethod* могут находиться следующие файлы:

- *payment.paymethod.php*
- *payment.paymethod.admin.php*
- *payment.paymethod.model.php*
- *payment.paymethod.view.php*

Это не обязательная структура платежного модуля. Нужно использовать только необходимые файлы. Кроме того, в платежный модуль можно добавлять свои файлы любой другой структуры.

Рассмотрим структуру стандартных файлов

payment.paymethod.php

Файл, который обрабатывает прямые запросы к платежному модулю. Обычно эти запросы указываются в настройках на сайте платежной системы. Прямой доступ к этому файлу идет по ссылке: *http://site.ru/payment/get/paymethod/*. Можно передать дополнительные аргументы, которые будут доступны через переменную `$_GET["rewrite"]`.

Пример:

```
// http://site.ru/payment/get/paymethod/success/
$_GET["rewrite"] = 'paymethod/success';

// http://site.ru/payment/get/paymethod/fail/
$_GET["rewrite"] = 'paymethod/fail';
```

Таким образом можно определять какое действие обрабатывать.

Чаще всего платежные системы предлагают два действия: успешное зачисление платежа (success) и неудачная попытка оплаты (fail). При этом передаются номер платежа (например, в переменной `$_REQUEST["id"]`) и какие-то данные для проверки валидности запроса (алгоритм предоставляется платежной системой).

Для начала нужно найти платеж в базе данных:

```
$pay = $this->diafan->_payment->check_pay($_REQUEST["id"], 'paymethod');
```

Если платеж не будет найден, то выполнение скрипта прервется и пользователь увидит ошибку 404.

Далее, если платеж неудачный, то вызываем функцию `fail()`.

```
$this->diafan->_payment->fail($pay);
```

Если платеж успешный, то выполняем валидацию данных по предоставленному системой алгоритму и зачисляем средства на счет.

```
if(проверка пройдена)
{
    $this->diafan->_payment->success($pay);
}
```

Настройки платежного метода, заданные администратором сайта доступны через переменную `$pay["params"]`.

Пример:

```
echo $pay["params"]["paymethod_login"];
// выведет настройку "Логин"
```

Бывают случаи, когда платежная система предлагает разбить зачисление успешного платежа на два шага: проверку и показ страницы об успешно окончании. Тогда зачисление средств надо проводить при проверке, а на втором этапе просто показать страницу успешного зачисления средств.

1 шаг:

```
if(проверка пройдена)
{
    $this->diafan->_payment->success($pay, 'pay');
}
```

2 шаг:

```
$this->diafan->_payment->success($pay, 'redirect');
```

payment.paymentmethod.admin.php

Файл содержит настройки платежного метода, подгружаемые при добавлении метода оплаты.

Структура файла:

Пример:

```
// ошибка 404 при прямом вызове файлов
if ( ! defined('DIAFAN'))
{
    $path = __FILE__; $i = 0;
    while(! file_exists($path.'/includes/404.php'))
    {
        if($i == 10) exit; $i++;
        $path = dirname($path);
    }
    include $path.'/includes/404.php';
}

class Payment_paymenthod_admin
{
    public $config;
    private $diafan;

    public function __construct(&$diafan)
    {
        $this->diafan = &$diafan;
        $this->config = array(
            "name" => 'Название метода, выводимое администратору',
            "params" => array(
                // название настройки должно начинаться с названия метода
                'paymenthod_string' => 'Название',

                // если настройка не строковая, то вместо названия указываем массив с
                // названием и типом
                // доступны следующие типы: text - строка, checkbox - галочка
                'paymenthod_checkbox' => array(
                    'name' => 'Тестовый режим',
                    'type' => 'checkbox'
                ),

                // можно определить свою функцию для вывода поля и его сохранения
                'paymenthod_var' => 'Название',
            )
        );
    }

    /**
     * Своя функция на редактирования настройки paymenthod_var
     *
     * @return void
     */
    public function edit_variable_paymenthod_var()
```

```

{
    echo '<div class="unit tr_payment" payment="paymenthod" style="display:none">
        <div class="infofield">Название поля</div>
        Значение поля
    </div>';
}

/**
 * Своя функция на сохранение настройки paymenthod_var
 *
 * @return void
 */
public function save_variable_paymenthod_var()
{
    // сохраняет данные
}
}

```

payment.paymenthod.model.php

Файл выводит данные платежной системы поле выбора метода оплаты. Чаще всего это форма оплаты, которая отправляет данные на сайт платежной системы. Также может быть прямой редирект на платежную систему или ссылки на платежные квитанции.

В файле генерируются данные, а затем они подставляются в шаблон *payment.paymenthod.view.php*.

Пример:

```

// ошибка 404 при прямом вызове файлов
if ( ! defined('DIAFAN'))
{
    $path = __FILE__; $i = 0;
    while(! file_exists($path.'/includes/404.php'))
    {
        if($i == 10) exit; $i++;
        $path = dirname($path);
    }
    include $path.'/includes/404.php';
}

class Payment_paymethod_model extends Diafan
{
    /**
     * Формирует данные для формы платежной системы
     *
     * @param array $params настройки платежной системы
     * @param array $pay данные о платеже
     * @return array
     */
}

```

```

public function get($params, $pay)
{
    $result = array(
        // какие-то данные, которые затем будут подставлены в шаблон
    );
    return $result;
}
}

```

Переменная `$params` — это массив настроек платежной системы, описанных в файле `payment.paymethod.admin.php`.

Пример:

```

// из нашего пример массив $params будет следующим
$params = array(
    'paymethod_string' => '...',
    'paymethod_checkbox' => 1|0,
    'paymethod_var' => '...',
);

```

Переменная `$pay` — это массив данных о платеже.

Пример:

```

$pay = array(
    'id' => 'идентификатор платежа из таблицы {payment_history}',
    'text' => '"Сообщение о совершенном заказе перед оплатой" из настроек модуля, в котором используется оплата, например, модуля "Магазин". Используется для вывода пользователю на экран.',
    'desc' => '"Описание платежа" из настроек модуля, в котором используется оплата, например, модуля "Магазин". Передается платежной системе.',
    // данные о плательщике, детали заказа
    'details' => array(
        'address' => 'Адрес',
        'street' => 'Улица',
        'building' => 'Номер дома',
        'suite' => 'Корпус',
        'flat' => 'Квартира',
        'entrance' => 'Подъезд',
        'floor' => 'Этаж',
        'intercom' => 'Домофон',
        'city' => 'Город',
        'country' => 'Страна',
        'zip' => 'Индекс',
        'metro' => 'Станция метро',
        'cargolift' => 'Наличие грузового лифта',
        'name' => 'ФИО',
        'firstname' => 'Имя',
        'lastname' => 'Фамилия',
    )
);

```



```

'fathersname' => 'Отчество',
'phone' => 'Телефон',
'phone-extra' => 'Дополнительный телефон',
'email' => 'Электронный адрес для связи',
'comment' => 'Комментарий к адресу',
// заказанные товары
'goods' => array(
    array(
        'name' => 'название',
        'article' => 'артикул',
        'price' => 'цена',
        'count' => 'количество заказанного товара',
        'summ' => 'сумма',
    ),
    ...
),
// сопутствующие услуги
'additional' => array(
    array(
        'name' => 'название',
        'summ' => 'сумма',
    ),
    ...
),
// доставка
'delivery' => array(
    'name' => 'название',
    'summ' => 'сумма',
),
'discount' => 'скидка на весь заказ (скидки на отдельные товары включены в цену товаров)',
),
);

```

Все детали заказа в массиве `$pay['details']` являются необязательными. Их наличие и заполнение зависит от модуля, использующего платежный модуль, а также от настроек модуля. Поэтому при использовании нужно проверять переменные на существование.

payment.paymentmethod.view.php

Шаблон платежного метода, который выводится в контентной части страницы после выбора метода оплаты. Данные для шаблона генерируются в файле `payment.paymentmethod.model.php`.

Модуль Webmoney

При подключении модуля WebMoney необходимо заполнить поле дополнительных настроек следующими значениями:

- Webmoney: секретный ключ – ключ, указанный в настройках системы Webmoney;

- Webmoney: кошелек – кошелек, выданный в системе Webmoney.

На сайте Webmoney необходимо внести в настройки следующие значения:

- ResultURL: **//yoursite.ru/payment/get/webmoney/result/**
- SuccessURL: **//yoursite.ru/payment/get/webmoney/success/**
- FailURL: **//yoursite.ru/payment/get/webmoney/fail/**

Модуль Баланс пользователя

Модуль не требует дополнительных настроек. Пользователь в любой момент в своем личном кабинете может пополнить баланс с помощью других методов оплаты, и затем, при выборе оплаты «Баланс пользователя» деньги будут списаны с баланса пользователя.

Модуль Банковские платежи

Служит для автоматической выписки счетов на оплату и квитанций. Необходимо заполнить дополнительные настройки модуля, реквизиты Вашей организации, от имени которой будут выписываться счета на оплату:

- 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 Вашей печати.

При выборе пользователем метода оплаты «Банковские платежи», для него будут сформированы платежные квитанции для физического лица и счета на оплату для юридического лица, в которые будут подставлены данные платежа и данные из настроек платежного модуля. Пользователь увидит их печатные формы, сможет распечатать и оплатить в любом банке.

Шаблоны квитанций:

- для физического лица –
modules/payment/backend/non_cash/payment.non_cash.fl.php
- для юридического лица –
modules/payment/backend/non_cash/payment.non_cash.ul.php

Модуль QIWI

В модуль QIWI на сайте необходимо внести следующие параметры, которые Вы получите на сайте QIWI при регистрации своего магазина:

- Номер терминала – ID магазина
- Пароль – пароль от API ID

На сайте QIWI необходимо внести следующие параметры:

- ResultURL: *http://yoursite.ru/payment/get/qiwi/result/*
- SuccessURL: *http://yoursite.ru/payment/get/qiwi/success/*

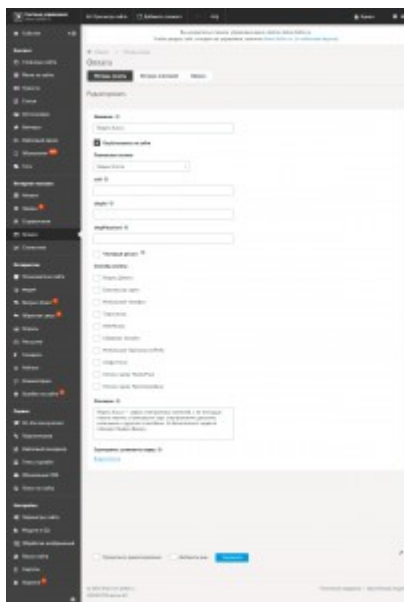
Модуль Яндекс.Касса

Для работы с этим модулем необходимо подключиться к Яндекс.Кассе и заключить договор с Яндекс.Деньгами, это могут сделать только юрлица или ИП. Настройки для активации модуля:

- scId - номер магазина в Яндексе. Выдается при подключении к Яндекс.Кассе.
- shopId - номер витрины в Яндексе. Выдается при подключении к Яндекс.Кассе.
- shopPassword - секретное слово, его можно указать в личном кабинете Яндекс.Кассы после подключения.
- Тестовый режим - включается для тестирования оплаты. Его нужно **выключить**, чтобы принимать настоящие платежи.
- Способы оплаты - выберите нужные (поставьте галочку). Доступные вам способы оплаты указаны в договоре с Яндекс.Деньгами.

Настройки на стороне Яндекс.Кассы (в личном кабинете):

- avisoURL, checkURL, checkOrder: *https://yoursite.ru/payment/get/yandexmoney/*
- successURL: *https://yoursite.ru/payment/get/yandexmoney/success/*
- failURL: *https://yoursite.ru/payment/get/yandexmoney/fail/*

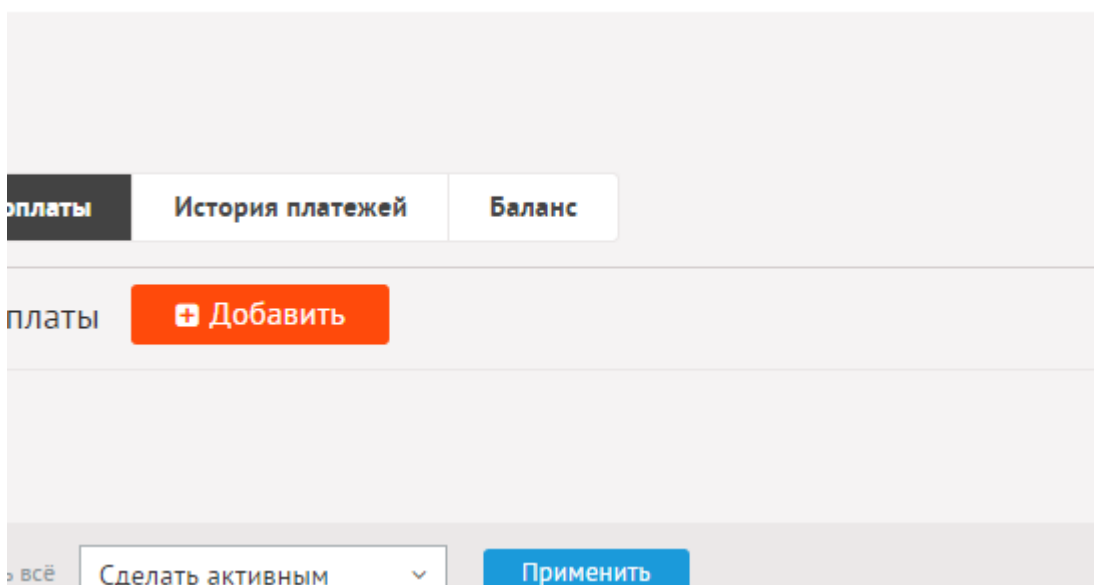


Платежные агрегаторы

В DIAFAN.CMS в стандартную сборку по умолчанию не включены многочисленные модули платежных агрегаторов (сервисов, предоставляющий возможность оплаты разными способами, включая банковские карты коммерческих банков). Их можно скачать и установить отдельно. [Список модулей агрегаторов](#)

Методы оплаты

Платежные модули (или бэкенды) подключаются к методам оплаты. А уже методы оплаты добавляются в отдельном интерфейсе, где для них определяется порядок следования, описание и подключенные платежные модули.



Методы оплаты имеют следующие свойства:

Основные:

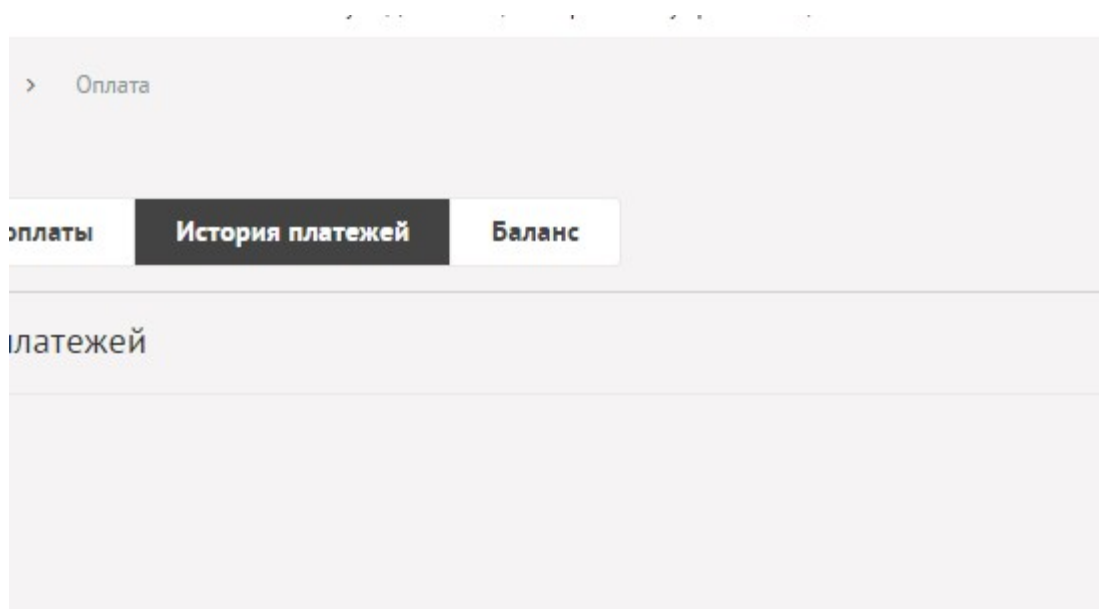
- **Название** – Название метода оплаты, выводится на сайте. Параметр имеет разные

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.
- **Описание** – Описание метода оплаты, выводится на сайте в форме заказа. Параметр имеет разные значения для разных языковых версий сайта.
- **Сортировка: установить перед** – Изменить положение текущего метода оплаты среди других методов. В списке методов можно сортировать методы простым перетаскиванием мыши.

История платежей

Просмотр истории платежей за заданный период времени.



База данных

[{payment}](#) – Методы оплаты

[{payment_history}](#) – История платежей

Файлы

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

1. *modules/payment/admin/js/payment.admin.js* – редактирование платежных систем, JS-сценарий;
2. *modules/payment/admin/payment.admin.php* – редактирование методов оплаты;
3. *modules/payment/admin/payment.admin.history.php* – история платежей;

4. *modules/payment/backend/balance/payment.balance.admin.php* – настройки платежной системы «Баланс пользователя» для административного интерфейса;
5. *modules/payment/backend/balance/payment.balance.model.php* – действия при выборе оплаты балансом;
6. *modules/payment/backend/balance/payment.balance.view.php* – шаблон вывода ошибки платежа при оплате балансом;
7. *modules/payment/backend/non_cash/payment.non_cash.php* – платежная квитанция на оплату;
8. *modules/payment/backend/non_cash/payment.non_cash.admin.php* – настройки платежной системы «Банковские платежи» для административного интерфейса;
9. *modules/payment/backend/non_cash/payment.non_cash.fl.php* – платежная квитанция на оплату для физ.лица;
10. *modules/payment/backend/non_cash/payment.non_cash.memo.php* – товарный чек;
11. *modules/payment/backend/non_cash/payment.non_cash.model.php* – формирует данные для страницы оплаты безналичным платежом;
12. *modules/payment/backend/non_cash/payment.non_cash.num2str.php* – перевод чисел в слова;
13. *modules/payment/backend/non_cash/payment.non_cash.ul.php* – платежная квитанция на оплату для юр.лица;
14. *modules/payment/backend/non_cash/payment.non_cash.view.php* – шаблон безналичного платежа;
15. *modules/payment/backend/non_cash/payment.non_cash.view.fl.php* – шаблон квитанции для физического лица;
16. *modules/payment/backend/non_cash/payment.non_cash.view.memo.php* – шаблон товарного чека;
17. *modules/payment/backend/non_cash/payment.non_cash.view.ul.php* – шаблон квитанции для юридического лица;
18. *modules/payment/backend/qiwi/payment.qiwi.php* – обработка данных, полученных от системы QIWI;
19. *modules/payment/backend/qiwi/payment.qiwi.admin.php* – настройки платежной системы QIWI для административного интерфейса;
20. *modules/payment/backend/qiwi/payment.qiwi.model.php* – формирует данные для формы платежной системы QIWI;

21. *modules/payment/backend/qiwi/payment.qiwi.view.php* – шаблон платежа через систему QIWI;
22. *modules/payment/backend/robokassa/payment.robokassa.php* – обработка данных, полученных от системы Robokassa;
23. *modules/payment/backend/robokassa/payment.robokassa.admin.php* – настройки платежной системы Robokassa для административного интерфейса;
24. *modules/payment/backend/robokassa/payment.robokassa.model.php* – формирует данные для формы платежной системы Robokassa;
25. *modules/payment/backend/webmoney/payment.webmoney.php* – обработка данных, полученных от системы WebMoney;
26. *modules/payment/backend/webmoney/payment.webmoney.admin.php* – настройки платежной системы WebMoney для административного интерфейса;
27. *modules/payment/backend/webmoney/payment.webmoney.model.php* – формирует данные для формы платежной системы WebMoney;
28. *modules/payment/backend/webmoney/payment.webmoney.view.php* – шаблон платежа через систему WebMoney;
29. *modules/payment/backend/yandexmoney/payment.yandexmoney.php* – работа с платежной системой Яндекс.Касса;
30. *modules/payment/backend/yandexmoney/payment.yandexmoney.admin.php* – настройки платежной системы Яндекс.Касса для административного интерфейса;
31. *modules/payment/backend/yandexmoney/payment.yandexmoney.model.php* – формирует данные для формы платежной системы Яндекс.Касса;
32. *modules/payment/backend/yandexmoney/payment.yandexmoney.view.php* – шаблон платежа через систему Яндекс.Касса;
33. *modules/payment/payment.get.php* – работа с платежными системами;
34. *modules/payment/payment.inc.php* – модель модуля «Методы оплаты»;
35. *modules/payment/payment.install.php* – установка модуля;
36. *modules/payment/views/payment.view.list.php* – шаблон списка платежных система при оплате.

Баланс пользователя

- [Подключение](#)
- [Баланс](#)
- [Настройки](#)
- [База данных](#)
- [Файлы](#)

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

Модуль позволяет прикрепить к пользователю баланс, пополнять его через платежные системы и расходовать на покупку товаров.

В административной части сайта модуль доступен в разделе «Оплата».

Подключение

Подключаемая часть – файл *modules/balance/balance.inc.php*. В нем описан класс *Balance_inc*. В модуле к объекту класса можно обратиться через переменную `$this->diafan->_balance`. Экземпляр класса создается при первом вызове переменной.

Методы

float **get** ([integer \$user_id = false]) – Возврат текущего баланса пользователя.

- integer \$user_id: ID пользователя. По умолчанию текущий пользователь

Пример:

```
echo 'Ваш баланс: '.$this->diafan->_balance->get().' руб.';
// выведет:
// Ваш баланс: 300 руб.
```

array **details** (integer \$user_id) – Возврат информации о плательщике.

- integer \$user_id: ID пользователя

void **set** (integer \$user_id, float \$summ, [string \$type = 'summ']) – Изменение текущего баланса пользователя.

- integer \$user_id: ID пользователя. По умолчанию текущий пользователь
- float \$summ: новая сумма
- string \$type: тип операции: *summ* – изменение всей суммы на балансе, *plus* – добавление, *minus* – вычитание

Пример:

```
// добавляет текущему пользователю 300 руб. на баланс
$this->diafan->_balance->set(0, 300, 'plus');
```

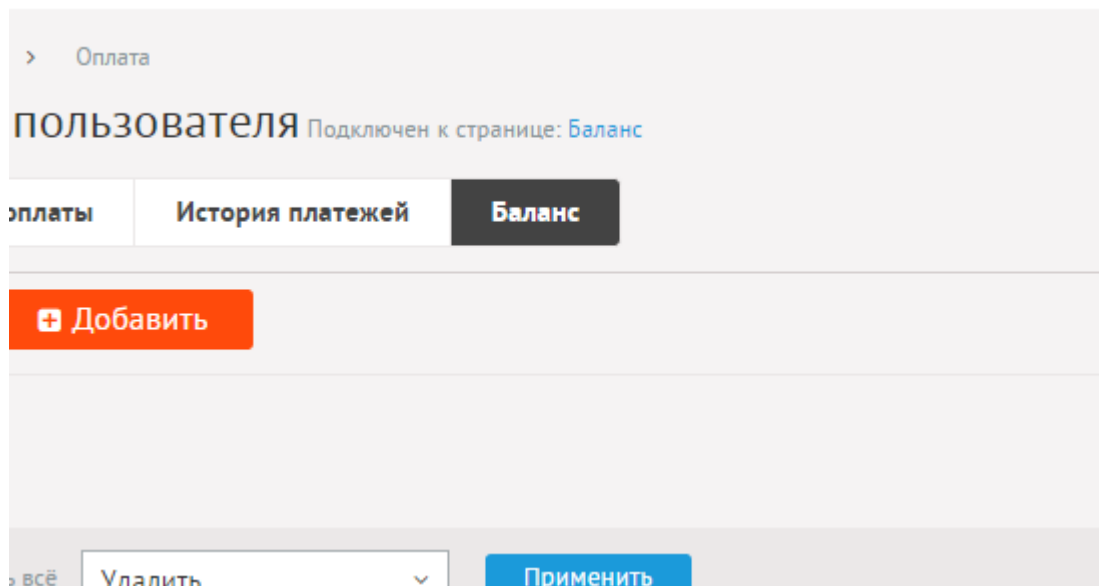
void **pay** (array \$pay) – Добавление средств на баланс пользователя.

- array \$pay: данные платежа

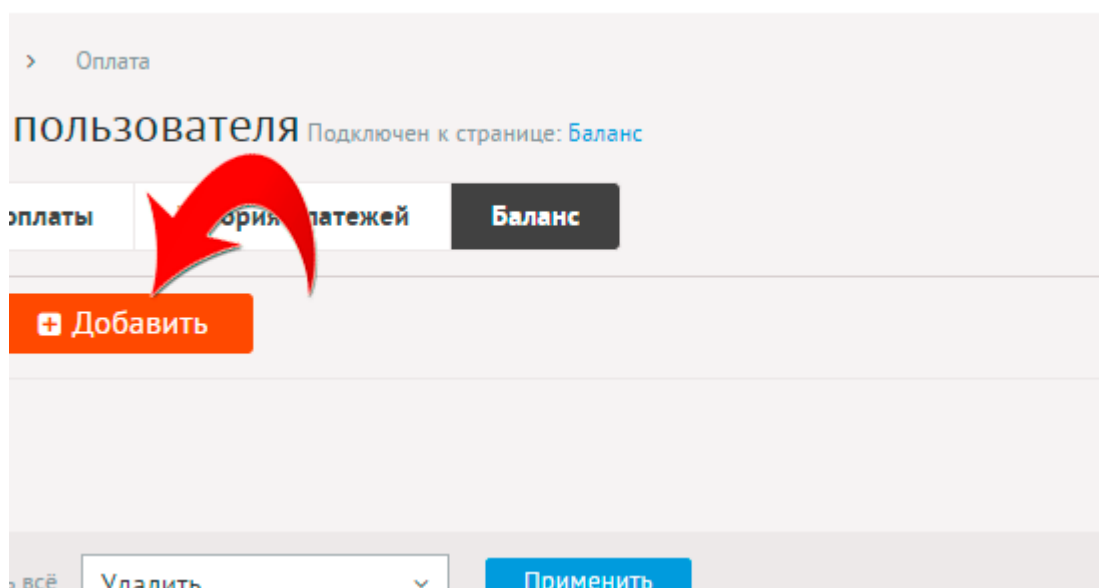
Пример:

```
// проводит пополнение баланса пользователя ID=4
$pay = DB::fetch_array("SELECT * FROM {payment_history} WHERE element_id=4 AND
module_name='balance'");
$this->diafan->_balance->pay($pay);
```

Баланс



Посмотреть и отредактировать баланс пользователей можно во вкладке «Баланс». Ссылка «Добавить» нужна для того, чтобы добавить средства на баланс пользователя, которого еще нет в списке. Записи появляются при пополнении баланса в пользовательской части.



При редактировании баланса можно задать следующие поля.

Основные:

- **Имя пользователя.**
- **Средства на балансе.**

Настройки

> Оплата

ПОЛЬЗОВАТЕЛЯ Подключен к странице: Баланс

оплаты История платежей Баланс

1

валюты ?

- **Название валюты** – Название валюты баланса. Параметр имеет разные значения для разных языковых версий сайта.
- **Сообщение о пополнении баланса перед оплатой** – Сообщение, получаемое пользователем по окончании пополнения баланса. Параметр имеет разные значения для разных языковых версий сайта.
- **Описание платежа** – Используется платежными системами. Можно добавлять:
 - %id – номер платежа.
 Параметр имеет разные значения для разных языковых версий сайта.
- **Платеж успешно принят** – Сообщение, которое увидит пользователь, если платеж успешно принят платежной системой.
- **Платеж не принят** – Сообщение, которое увидит пользователь, если платеж не принят платежной системой.
- **Тема письма пользователю о пополнении баланса** – Можно добавлять:
 - %title – название сайта,
 - %url – адрес сайта (например, site.ru),
 - %id – номер платежа.
 Параметр имеет разные значения для разных языковых версий сайта.
- **Сообщение пользователю о пополнении баланса** – Можно добавлять:
 - %title – название сайта,
 - %url – адрес сайта (например, site.ru),
 - %payment – способ оплаты,
 - %fio – имя пользователя,
 - %id – номер платежа.
 Параметр имеет разные значения для разных языковых версий сайта.
- **Тема письма администратору о пополнении баланса** – Можно добавлять:
 - %title – название сайта,

- %url – адрес сайта (например, site.ru),
- %id – номер платежа.
- **Текст письма администратору о пополнении баланса** – Можно добавлять:
 - %title – название сайта,
 - %url – адрес сайта (например, site.ru),
 - %payment – способ оплаты,
 - %fio – имя пользователя, пополнившего баланс,
 - %id – номер платежа.
- **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 пополнении баланса по SMS** – Возможность отправлять SMS администратору при пополнении баланса. Параметр можно подключить, если в [Параметрах сайта](#) настроены SMS-уведомления.
- **Номер телефона в федеральном формате** – Номер телефона для SMS-уведомлений администратора о пополнении баланса. Параметр появляется, если отмечена опция «Уведомлять о пополнении баланса по SMS».
- **Сообщение для уведомлений** – Текст сообщения для SMS-уведомлений администратора о пополнении баланса. Не более 800 символов. Параметр появляется, если отмечена опция «Уведомлять о пополнении баланса по SMS».

База данных

[{balance}](#) – Баланс пользователей

Файлы

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

1. *modules/balance/admin/balance.admin.php* – редактирование баланса пользователей;
2. *modules/balance/admin/balance.admin.config.php* – настройки модуля;
3. *modules/balance/balance.php* – контроллер модуля «Баланс пользователя»;
4. *modules/balance/balance.action.php* – обрабатывает полученные данные из формы;
5. *modules/balance/balance.inc.php* – подключение модуля «Баланс пользователя»;
6. *modules/balance/balance.install.php* – установка модуля;
7. *modules/balance/balance.model.php* – модель модуля «Баланс пользователя»;
8. *modules/balance/views/balance.view.form.php* – шаблон формы пополнения баланса пользователя;
9. *modules/balance/views/balance.view.payment.php* – шаблон формы платежной системы;

10. *modules/balance/views/balance.view.result.php* – шаблон подтверждения/опровержения платежа.

Геокарта

- [Подключение](#)
- [Структура бэкенда](#)
- [Как добавлять геокарта к модулю](#)
- [Административная часть](#)
- [База данных](#)
- [Файлы](#)

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

Модуль «Геокарта» подключается к другим модулям. В стандартной комплектации DIAFAN.CMS «Геокарта» подключен к модулю «Объявления». В рамках модуля карта подключается в настройках конкретного модуля.

Модуль позволяет отмечать точки на карте для элементов другого модуля. Одному элементу соответствует одна точка. Если подключить модуль к странице сайта, то можно вывести все точки на одной карте. GET-переменная *module* выводит точки только для выбранного модуля.

Пример:

Модуль «Геокарта» подключен к модулям «Статьи» и «Объявления». Создана страница «Геокарта» <http://site.ru/geomap/>. На этой странице будет выведена карта с точками всех модулей: и статей, и объявлений. Страница <http://site.ru/geomap/?module=ab> покажет только точки объявлений.

Подключение

Подключаемая часть – файл *modules/geomap/geomap.inc.php*. В нем описан класс *Geomap_inc*. В модуле к объекту класса можно обратиться через переменную `$this->diafan->_geomap`. Экземпляр класса создается при первом вызове переменной.

Методы

`string get ([integer $element_id = 0], [string $module_name = ''], [string $element_type = 'element'], [integer $site_id = 0])` – Показывает точку на карте для элемента.

- `integer $element_id`: номер элемента модуля, по умолчанию текущий элемент модуля
- `string $module_name`: название модуля, по умолчанию текущий модуль
- `string $element_type`: тип данных
- `integer $site_id`: страница сайта, к которой прикреплен элемент, по умолчанию текущая страница сайта

Для того, чтобы вывести точку на карте в карточке элемента на сайте, нужно в шаблоне модуля *modules/модуль/views/модуль.view.id.php* добавить:

Пример:

```
echo $this->diafan->_geomap->get($result["id"], 'модуль');
```

void **prepare** ([integer \$element_id = 0], [string \$module_name = "], [string \$element_type = 'element']) – Запоминает данные элемента, которому нужно будет вывести точки на карте.

- integer \$element_id: номер элемента модуля, по умолчанию текущий элемент модуля
- string \$module_name: название модуля, по умолчанию текущий модуль
- string \$element_type: тип данных

string **add** ([integer \$element_id = 0], [string \$module_name = "], [string \$element_type = 'element'], [integer \$site_id = 0]) – Редактирование/добавление точки на карте.

- integer \$element_id: номер элемента модуля при редактировании точки
- string \$module_name: название модуля при редактировании точки
- string \$element_type: тип данных
- integer \$site_id: страница сайта, к которой прикреплен элемент, по умолчанию текущая страница сайта

Для вывода карты в форме добавления элемента, нужно в шаблон формы добавления (обычно это файл *modules/модуль/views/модуль.view.form.php*) включить код:

Пример:

```
echo $this->diafan->_geomap->add(0, 'модуль');
```

Редактирование точки на карте для элемента подключается в шаблоне редактирования элемента (обычно это файл *modules/модуль/views/модуль.view.edit.php*) следующим кодом:

Пример:

```
echo $this->diafan->_geomap->add($result["id"], 'модуль');
```

string **save** ([integer \$element_id = 0], [string \$module_name = "], [string \$element_type = 'element'], [integer \$site_id = 0]) – Сохранение точки на карте.

- integer \$element_id: номер элемента модуля
- string \$module_name: название модуля
- string \$element_type: тип данных
- integer \$site_id: страница сайта

Для сохранения данных карты в функцию сохранения нужно включить код:

```
$this->diafan->_geomap->save($id, "модуль");
```

Пример:

В модуле «Объявления» обработка запросов на добавление и сохранение объявления происходит в файле *modules/ab/ab.action.php*. В функцию добавления объявления `add()` после SQL-запроса на сохранение объявления включен код:

```
$this->diafan->_geomap->save($save, "ab");
```

В функцию сохранения отредактированного объявления `save()` добавлен код:

```
$this->diafan->_geomap->save($_POST["id"], "ab");
```

void delete (integer|array \$element_ids, string \$module_name, [string \$element_type = 'element']) – Удаляет точки для одного или нескольких элементов.

- integer|array \$element_ids: номер одного или нескольких элементов
- string \$module_name: название модуля
- string \$element_type: тип данных

Пример:

В модуле «Объявления» в функцию удаления объявления `delete()` в файле *modules/ab/ab.action.php* включен следующий код:

```
$this->diafan->_geomap->delete($row["id"], "ab");
```

void delete_module (string \$module_name) – Удаляет все точки элементов модуля.

- string \$module_name: название модуля

Пример:

```
// удалит точки на карте всех объявлений
$this->diafan->_geomap->delete_module('ab');
```

array config () – Настройки бэкенда.

Структура бэкенда

К модулю можно подключить неограниченное количество бэкендов – модулей для работы с конкретными картами.

Для этого нужно разработать несколько файлов стандартной структуры, учитывающие особенности API карт и положить папку с этими файлами в папку *geomap/backend*. Бэкенд появится в административной части модуля «Геокарта» и будет доступен к выбору.

Бэкенд должен иметь уникальное название латинскими буквами (по примеру *yandex*, *google*, *2gis*). Для примера будем использовать название *newmap*.

В папке *geomap/backend/newmap* должны находиться следующие файлы:

- *geomap.newmap.admin.php*
- *geomap.newmap.view.add.php*
- *geomap.newmap.view.get.php*
- *geomap.newmap.view.get_all.php*

Рассмотрим структуру стандартных файлов

geomap.newmap.admin.php

Файл содержит название и настройки бэкенда, подгружаемые при выборе его из списка.

Структура файла:

Пример:

```
// ошибка 404 при прямом вызове файлов
if ( ! defined('DIAFAN'))
{
    $path = __FILE__; $i = 0;
    while(! file_exists($path.'/includes/404.php'))
    {
        if($i == 10) exit; $i++;
        $path = dirname($path);
    }
    include $path.'/includes/404.php';
}

class Geomap_newmap_admin
{
    public $config;
    private $diafan;

    public function __construct(&$diafan)
    {
        $this->diafan = &$diafan;
        $this->config = array(
            "name" => 'Название, выводимое администратору',
            "params" => array(
                // название настройки должно начинаться с названия метода
                'newmap_string' => 'Название',

                // если настройка не строковая, то вместо названия указываем массив с
                названием и типом
                // доступны следующие типы: text - строка, checkbox - галочка
                'newmap_checkbox' => array(
                    'name' => 'Тестовый режим',
                    'type' => 'checkbox'
                ),

                // можно определить свою функцию для вывода поля и его сохранения
                'newmap_var' => 'Название',
            )
        );
    }

    /**
     * Своя функция на редактирования настройки newmap_var
     */
}
```



```

* @return void
*/
public function edit_variable_newmap_var()
{
    echo '<div class="unit tr_geomap" backend="paymentmethod" style="display:none">
        <div class="infofield">Название поля</div>
        Значение поля
    </div>';
}

/**
 * Своя функция на сохранение настройки newmap_var
 *
 * @return void
 */
public function save_variable_newmap_var()
{
    // сохраняет данные
}
}

```

gemap.newmap.view.add.php

Шаблон редактирования точки на карте.

В шаблоне доступны следующие данные:

- `$result["point"]` – координаты точки.
- `$result["config"]` – массив с настройками бэкенда.

gemap.newmap.view.get.php

Шаблон вывода точки на карте без возможности редактирования.

В шаблоне доступны следующие данные:

- `$result["point"]` – координаты точки.
- `$result["config"]` – массив с настройками бэкенда.

gemap.newmap.view.get_all.php

Шаблон вывода нескольких точек на карте.

В шаблоне доступны следующие данные:

- `$result["rows"]` – массив точек.
- `$result["config"]` – массив с настройками бэкенда.

Как добавлять геокарта 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 **'geomap'**:

```

public $variables = array(
    'main' => array(
        'geomap' => 'module',
        ...
    ),
    ...
);

```

Административная часть

В административной части модуля можно выбрать бэкенд – сервис, используемый для отображения карты. В стандартную сборку DIAFAN.CMS включены бэкенды «Яндекс.Карты» и «Google.Мap».

Каждый бэкенд может иметь свои настройки. Например, для Яндекс.Карты настраивается центр карты и масштаб.

База данных

[{geomap}](#) – Точки на карте

Файлы

1. *modules/geomap/admin/geomap.admin.php* – настройки модуля;
2. *modules/geomap/admin/geomap.admin.inc.php* – подключение модуля к административной части других модулей;
3. *modules/geomap/admin/js/geomap.admin.js* – редактирование бэкенда для геокарты, JS-сценарий;
4. *modules/geomap/backend/google/geomap.google.admin.php* – настройки карты "Google Maps" для административного интерфейса;
5. *modules/geomap/backend/google/geomap.google.view.add.php* – шаблон редактирования точки на карте "Google Maps";
6. *modules/geomap/backend/google/geomap.google.view.get.php* – шаблон точки на карте "Google Maps";
7. *modules/geomap/backend/google/geomap.google.view.get_all.php* – шаблон вывода нескольких точек на карте "Google Maps";
8. *modules/geomap/backend/yandex/geomap.yandex.admin.php* – настройки карты «Яндекс.Карты» для административного интерфейса;
9. *modules/geomap/backend/yandex/geomap.yandex.view.add.php* – шаблон редактирования точки на карте Яндекс.Карты;
10. *modules/geomap/backend/yandex/geomap.yandex.view.get.php* – шаблон точки на

карте Яндекс.Карты;

11. *modules/geomap/backend/yandex/geomap.yandex.view.get_all.php* – шаблон вывода нескольких точек на карте Яндекс.Карты;

12. *modules/geomap/geomap.php* – контроллер;

13. *modules/geomap/geomap.inc.php* – подключение модуля;

14. *modules/geomap/geomap.install.php* – установка модуля;

15. *modules/geomap/geomap.model.php* – модель;

16. *modules/geomap/views/geomap.view.show.php* – шаблон вывода нескольких точек на карте.

Изображения

- [Подключение](#)
- [Как добавить прикрепленные изображения к модулю](#)
- [Размеры изображений](#)
- [Настройки](#)
- [База данных](#)
- [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, string \$element_type, integer \$site_id, string \$alt, [integer \$param_id = 0], [integer \$count = 0], [string \$link_to = "], [string \$tmpcode = "])** – Получает изображения, прикрепленные к элементу модуля.

- 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
- string \$element_type: тип данных (*element* – элемент (по умолчанию), *cat* – категория, *brand* – производитель)
- integer \$site_id: страница сайта, к которой прикреплен элемент
- string \$alt: альтернативный текст получаемых изображений
- integer \$param_id: номер параметра, к которому прикреплено изображение
- integer \$count: количество изображений
- string \$link_to: размер изображения, на который ведет ссылка
- string \$tmpcode: временный идентификационный код элемента, к которому прикреплены 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", "element", $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 $img)
{
    echo '<a href="'.BASE_PATH_HREF.$img["link"]."'>'
    ''
    '</a> ';
}
```

Все размеры изображения содержатся в массиве `$img["vs"]` для каждого изображения в цикле. Ключами массива являются теги размера, указанные в настройках модуля.

Пример:

```
// выводим изображения с тегом preview
foreach($images as $img)
{
    echo '';
}
```

Рассмотрим как это можно использовать в шаблоне модуля.

Пример:

В настройках модуля «Интернет магазин» для товаров в списке размеров изображения («Генерировать размеры изображений») есть три созданные системой размера, помеченные тегами "large", "medium", "preview". Веб-мастер создал дополнительный размер изображения в интерфейсе «Обработка изображений». В настройках модуля он добавил этот размер и задал ему тег "new". Настройки применены к уже загруженным изображениям.

Теперь в шаблоне карточки товара `modules/shop/views/shop.view.id.php` он может обратиться к новому размеру изображения следующим образом:

```
// цикл по всем загруженным изображениям
foreach ($result["img"] as $img)
{
    // выводит изображение, помеченное тегом "medium"
    echo '';

    // выводит изображение, помеченное тегом "large"
    echo '';

    // выводит изображение, помеченное тегом "new"
    echo '';
}
```

Как видно из примера размер изображения (ширина "width" и высота "height") определяются только для изображения, помеченного тегом "medium". Если Вы зададите

эти атрибуты для любого другого размера изображения, то изображение просто будет растянуто или сжато под указанные размеры браузером, что приведет к его искаженному отображению.

В данном примере переменная `$img["link"]` содержит ссылку на размер, помеченный тегом "large". В списке товаров эта переменная может содержать ссылку на карточку товара (если выведены не все изображения). Поэтому ссылка относительная и ее следует использовать вместе с константами `BASE_PATH` или `BASE_PATH_HREF`.

Во всех остальных шаблонах модулей данные о изображениях представлены в этом же формате. Если выводиться карточка элемента, то изображения записаны в переменную `$result["img"]`. Если изображения выводятся в списке элементов, то `$result` заменяем на название переменной, которая содержит данные об элементе в цикле. Например, для списка товаров это переменная `$row`. Для списка категорий это переменная `$cat`, для списка подкатегорий это переменная `$child`.

Именно в модуле «Интернет магазин» в карточке товара есть еще переменная `$img["preview"]`, которая содержит ссылку на изображения размера "preview", если он есть в настройках модуля. В этом случае можно еще добавить вывод этого размера:

```
// если в настройках задан размер изображения, помеченный тегом "preview"
if($result["preview_images"])
{
    // цикл по всем загруженным изображениям
    foreach ($result["img"] as $img)
    {
        // выводит изображение, помеченное тегом "preview"
        echo '';
    }
}
```

`array prepare (integer $element_id, string $module_name, [string $element_type = 'element'], [integer $param_id = 0])` – Запоминает данные об элементах, которым нужно вывести прикрепленные изображения.

- integer \$element_id: номер элемента, к которому прикреплены изображения
- string \$module_name: название модуля, по умолчанию модуль, прикрепленный к текущей странице
- string \$element_type: тип данных (*element* – элемент (по умолчанию), *cat* – категория, *brand* – производитель)
- integer \$param_id: номер параметра

Пример:

```
// в этом примере будет выполнено три SQL-запроса к базе данных для получения
изображений ко всем заданным объявлениям
$ids = array(3, 5, 7);
foreach($ids as $id)
{
```

```
$images[$id] = $this->diafan->_images->get('medium', $id, 'ab', 'element', 5, "Объявление", false, 2, "large");
}
```

Пример:

```
// в этом примере будет выполнен один SQL-запрос к базе данных для получения
// изображений ко всем заданным объявлениям
$ids = array(3, 5, 7);
foreach($ids as $id)
{
    $this->diafan->_images->prepare($id, 'ab');
}
foreach($ids as $id)
{
    $images[$id] = $this->diafan->_images->get('medium', $id, 'ab', 'element', 5, "Объявление", false, 2, "large");
}
```

void delete (integer|array \$element_ids, string \$module_name, [string \$element_type = 'element'], [integer \$param_id = false]) – Удаляет прикрепленные изображения.

- integer|array \$element_ids: номер одного или нескольких элементов
- string \$module_name: название модуля
- string \$element_type: тип данных (*element* – элемент (по умолчанию), *cat* – категория, *brand* – производитель)
- integer \$param_id: номер дополнительной характеристики с типом «Изображения»

Пример:

```
//удаляем все изображения, прикрепленные к категории ID=3 в файловом архиве
$this->diafan->_images->delete(3, "files", "cat");

//удаляем все изображения, прикрепленные к категориям ID=3,5,6 в файловом архиве
$this->diafan->_images->delete(array(3, 5, 6), "files", "cat");
```

void delete_row (array \$row) – Удаляет одно изображение.

- array \$row: информация о изображении, записанная в базу данных

Пример:

```
// получаем данные об изображении ID=3
$row = DB::query_fetch_array("SELECT * FROM {images} WHERE id=3");

//удаляем изображение
$this->diafan->_images->delete_row($row);
```

integer copy_row (integer \$id, [integer \$element_id = false], [string \$tmpcode = false]) – Копирует одно изображение.

- integer \$id: номер записи в таблице
- integer \$element_id: номер элемента, к которому прикреплены изображения
- string \$tmpcode: временный идентификационный код элемента, к которому прикреплены изображения

void **delete_module (string \$module_name)** – Удаляет все изображения модуля.

- string \$module_name: название модуля

Пример:

```
// удалит изображения, прикрепленные ко всем объявлениям и категориям объявлений
$this->diafan->_images->delete_module('ab');
```

mixed **upload (integer \$element_id, string \$module_name, string \$element_type, integer \$site_id, string \$tmpfile, string \$new_name, [boolean \$handle = false], [integer \$param_id = 0], [string \$tmpcode = ""])** – Загружает прикрепленные изображения.

- integer \$element_id: номер элемента, к которому прикреплены изображения
- string \$module_name: название модуля
- string \$element_type: тип данных (*element* – элемент (по умолчанию), *cat* – категория, *brand* – производитель)
- integer \$site_id: страница сайта
- string \$tmpfile: расположение файла
- string \$new_name: название файла без расширения
- boolean \$handle: ручная загрузка изображений по одной
- integer \$param_id: номер дополнительной характеристики с типом «Изображения»
- string \$tmpcode: временный код для прикрепления изображений к еще не созданному элементу

Пример:

```
try
{
    // загружаем изображение для производителя ID=5 в магазине, подключенном к
    // странице ID=3
    $this->diafan->_images->upload(5, 'shop', 'brand', 3, $_FILES["images"], 'proizvoditel_oop');
}
catch(Exception $e)
{
    // сообщение об ошибке при загрузке
    echo $e->getMessage();
}
```

void **get_variation_image (string \$file_name, string \$module_name, array \$variation, integer \$folder_num, [boolean \$handle = false], [boolean \$after_selectarea = false])** – Применяет вариант к изображению.

- string \$file_name: название файла
- string \$module_name: название модуля

- array \$variation: данные о варианте
- integer \$folder_num: номер папки
- boolean \$handle: ручная обработка (для вырезания части изображения)
- boolean \$after_selectarea: обработка после ручного выделения области

Пример:

```
// получаем данные о варианте изображения, который нужно применить к
// изображению
$variation = DB::query_fetch_array("SELECT * FROM {images_variations} WHERE id=5");

// генерируем копию изображения в полученном размере
$this->diafan->_images->get_variation_image("izobrazhenie_tovara.jpg", "shop", $variation,
0);
```

integer **id_hash (string \$hash, string \$module_name)** – Определяет номер записи, которой соответствует хэш.

- string \$hash: хэш
- string \$module_name: название модуля

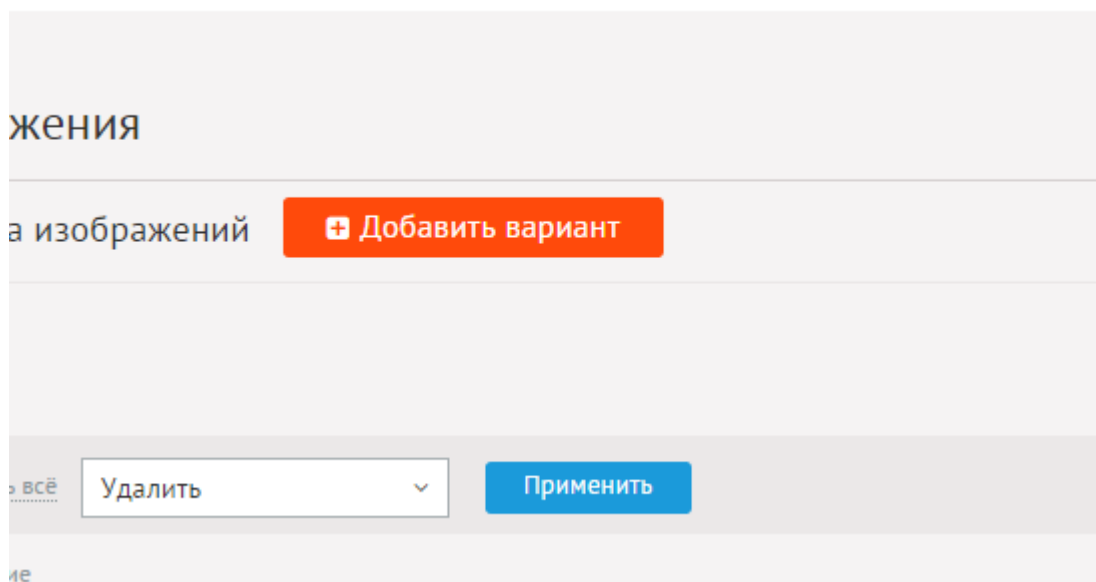
Как добавить прикрепленные изображения к модулю

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'**:

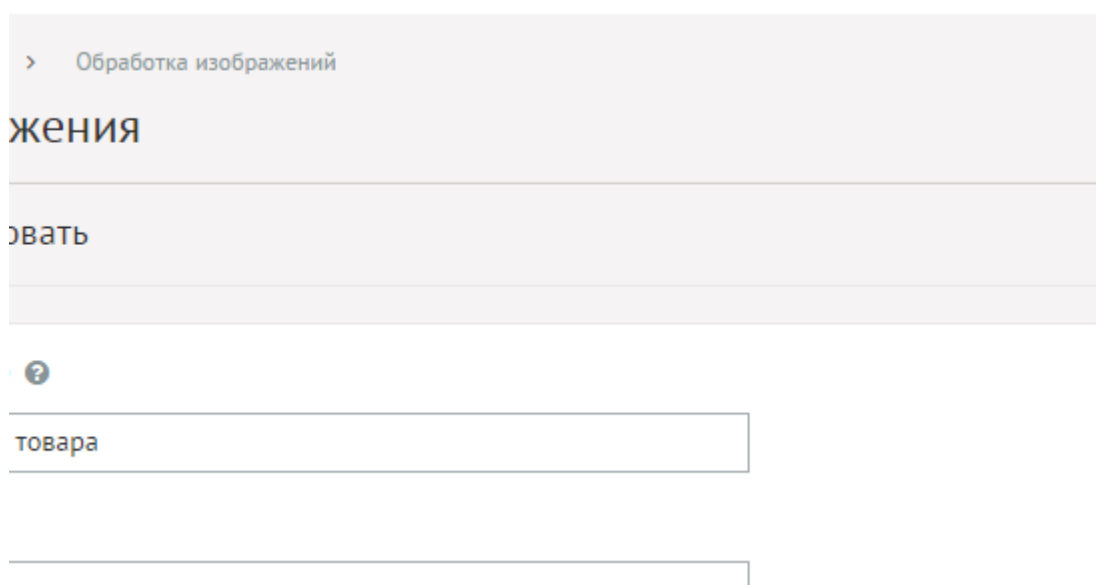
Пример:

```
public $variables = array(
    'main' => array(
        'images' => 'module',
        ...
    ),
    ...
);
```

Размеры изображений



В административной части модуля определяются варианты изображений, которые затем используются в настройках модуля и настройках визуального редактора.



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 (tag medium) и полное изображение (tag 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.

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».

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».

База данных

[`{images}`](#) – Прикрепленные изображения

[`{images_editor_folders}`](#) – Папки изображений в плагине для визуального редактора

[`{images_variations}`](#) – Варианты загрузки изображений

Файлы

1. `modules/images/admin/images.admin.php` – редактирование вариантов генерирования изображений;
2. `modules/images/admin/images.admin.action.php` – обработка POST-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.view.php` – шаблон вывода изображений в административной части;

5. *modules/images/admin/js/images.admin.inc.config.js* – подключение модуля к настройкам других модулей, JS-сценарий;
6. *modules/images/admin/js/images.admin.inc.js* – подключение модуля к административной части других модулей, JS-сценарий;
7. *modules/images/admin/js/images.admin.js* – редактирование вариантов генерирования изображений, JS-сценарий;
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.inc.php* – подключение для работы с прикрепленными изображениями;
11. *modules/images/images.install.php* – установка модуля;
12. *modules/images/js/images.editor.js* – плагин для визуального редактора, JS-сценарий.

Прикрепленные файлы

- [Подключение](#)
- [Как добавить прикрепленные файлы к модулю](#)
- [Настройки](#)
- [База данных](#)
- [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 \$param_id = 0])** – Отдает информацию о прикрепленных файлах.

- integer \$element_id: номер элемента, к которому прикрепляется файл
- string \$module_name: название модуля
- 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"]}.)'
    . ' '. $a_href2. <img src=${att["link_preview"]}></a></p>;
}
else
{
    echo '<p id="attachment".${att["id"]}><a href=${att["link"]}>'.
    . ${att["name"]}.</a> ('. ${att["size"]}.)</p>;
}
}
}

```

void **prepare** (integer \$element_id, string \$module_name, [integer \$param_id = 0]) –

Запоминает данные элемента, которому нужно вывести информацию о прикрепленных файлах.

- integer \$element_id: номер элемента, к которому прикрепляется файл
- string \$module_name: название модуля
- integer \$param_id: номер параметра, к которому прикреплен файл

Пример:

```

// в этом примере будет выполнено три SQL-запроса к базе данных для получения
// файлов ко всем заданным вопросам
$ids = array(3, 5, 7);
foreach($ids as $id)
{
    $files[$id] = $this->diafan->_attachments->get($id, 'faq');
}

```

Пример:

```

// в этом примере будет выполнен один SQL-запрос к базе данных для получения
// файлов ко всем заданным вопросам
$ids = array(3, 5, 7);
foreach($ids as $id)
{
    $this->diafan->_attachments->prepare($id, 'faq');
}
foreach($ids as $id)
{
    $files[$id] = $this->diafan->_attachments->get($id, 'faq');
}

```

boolean **save** (integer \$element_id, string \$module_name, [array \$config = array()]) – Сохраняет добавленные файлы.

- integer \$element_id: номер элемента

- string \$module_name: название модуля
- array \$config: конфигурация

Пример:

```
// загружаем файлы, прикрепленные к задаваемому вопросу
// в файле modules/faq/faq.action.php
$config = array('site_id' => $site_id, 'type' => 'configmodules');
try
{
    // загружаем файл для вопроса ID=3 в модуле «Вопрос-Ответ»
    $this->diafan->_attachments->save(3, "faq", $config);
}
catch(Exception $e)
{
    // сообщение об ошибке при загрузке
    echo $e->getMessage();
}
```

boolean **upload** (array \$file, string \$module_name, integer \$element_id, [integer|boolean \$n = false], [array \$config = array()]) – Загружает файлы.

- array \$file: загружаемый файл/файлы
- string \$module_name: название модуля
- integer \$element_id: номер элемента, к которому прикрепляется файл
- integer|boolean \$n: номер файла в массиве файлов, если предан массив
- array \$config: конфигурация

Пример:

```
try
{
    // загружаем файл для вопроса ID=3 в модуле «Вопрос-Ответ»
    $this->diafan->_attachments->upload($_FILES['attachment'], 'faq', 3);
}
catch(Exception $e)
{
    // сообщение об ошибке при загрузке
    echo $e->getMessage();
}
```

void **delete** (integer|array \$element_ids, string \$module_name, [integer \$attachment_id = 0], [integer \$param_id = 0]) – Удаляет прикрепленные файлы/файл.

- integer|array \$element_ids: номер одного или нескольких элементов
- string \$module_name: название модуля
- integer \$attachment_id: номер файла
- integer \$param_id: номер параметра, к которому прикреплен файл

Пример:


```
//удаляем файлы, прикрепленные к вопросу в модуле «Вопрос-Ответ»
$this->diafan->_attachments->delete(3, "faq");

//удаляем файлы, прикрепленные к вопросам ID=3,5,6 в модуле «Вопрос-Ответ»
$this->diafan->_attachments->delete(array(3, 5, 6), "faq");
```

void **delete_module (string \$module_name)** – Удаляет все прикрепленные файлы модуля.

- string \$module_name: название модуля

Пример:

```
// удалит файлы, прикрепленные ко всем вопросам в модуле «Вопрос-Ответ»
$this->diafan->_attachments->delete_module('faq');
```

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

База данных

[{attachments}](#) – Прикрепленные файлы

Файлы

1. *modules/attachments/admin/attachments.admin.inc.php* – подключение модуля к административной части других модулей;
2. *modules/attachments/admin/js/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;
5. *modules/attachments/attachments.install.php* – установка модуля.

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->diafan->_paginator`. Экземпляр класса создается при первом вызове переменной.

Свойства

var **nastr** – количество элементов, показанных на странице.

Пример:

```
// определение переменной в коде программы имеет больший приоритет,  
// чем определение соответствующего поля в настройках модуля  
$this->diafan->_paginator->nastr = 30;
```

var **variable** = 'page' – название переменной, содержащей номер страницы.

Если одновременно нужно подключить две постраничные навигации, то задаем свойство **variable** для дополнительной навигации отличное от *page*.

Пример:

Комментарии, подключенные к категории новостей не могу быть разбиты на странице с помощью переменной *page* (<http://site.ru/news/page2/>), так как эта переменная используется постраничной навигации новостей в категории. Поэтому используется дополнительная переменная *dpage* (<http://site.ru/news/dpage2/>).

```
$this->diafan->_paginator->variable = 'dpage';
```

var **navnastr** = 10 – количество ссылок постраничной навигации, показанных на одной странице.

Пример:

```
// сокращаем количество ссылок в постраничной навигации  
$this->diafan->_paginator->navnastr = 3;  
  
// выведет:  
// 1 2 3 ...» или « 6 7 8 »  
// вместо  
// 1 2 3 4 5 6 7 8 9 10 »
```

var **navlink** – ссылка на первую страницу.

Пример:

```
// адрес первой страницы счетчика просмотров фотографий
$this->diafan->_paginator->navlink = 'photo/counter/'.($this->diafan->_route->cat ? 'cat'.
$this->diafan->_route->cat.'/': '');
```

var **page** – номер страницы.

Пример:

```
// в особых случаях постраничная навигации можно использовать свою переменную для
постраничной навигации, заданную в виде GET переменной
$this->diafan->_paginator->page = !empty($_GET["page"]) ? intval($_GET["page"]) : 0;
```

var **last_page** – номер последней страницы.

var **get_nav** – GET переменные, участвующие в навигации для постраничного вывода.

Пример:

```
// добавляем ко всем ссылкам постраничной навигации
// GET-переменную с поисковым словом
if(! empty($_GET["searchword"]))
{
    $this->diafan->_paginator->get_nav =
        '?searchword='.$this->diafan->filter($_GET, "url", "searchword");
}
```

var **urlpage** = 'page%d/' – шаблон части ссылки, отвечающей за передачу номера страницы.

Пример:

```
// вместо переменной &nbsnavlink_tpl; можно задать urlpage, тогда navlink_tpl будет
равна navlink + urlpage
$this->diafan->_paginator->urlpage = 'page%d/';
```

var **polog** – порядковый номер элемента, с которого начинается вывод элементов.

Пример:

```
// получаем список ссылок в постраничной навигации
$links = $this->diafan->_paginator->get();

// 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);
```

var **nen** – количество элементов в списке.

Пример:

```
// количество новостей
$this->diafan->_paginator->nen = DB::query_result("SELECT COUNT(*) FROM {news}");
```

var **show_more** = false – ссылка "Показать ещё".

Методы

array **get** ([string \$module = false], [string \$action = false], [array \$attributes = false]) –
Формирует строку навигации.

- string \$module: имя модуля, обрабатывающего action события show_more
- string \$action: имя функции, обрабатывающей action события show_more
- array \$attributes: аргументы функции, обрабатывающей action события show_more

Пример:

```
// получаем список ссылок в постраничной навигации
$links = $this->diafan->_paginator->get();
```

```
// подставляем полученные данные в шаблон
$this->result["paginator"] = $this->diafan->_tpl->get('get', 'paginator', $links);
```

Самый простой пример использования постраничной навигации.

Пример:

```
// общее количество новостей
$this->diafan->_paginator->nen = DB::query_result("SELECT COUNT(*) FROM {news}");
```

```
// постраничная навигация
$this->result["paginator"] = $this->diafan->_tpl->get('get', 'paginator', $this->diafan->_paginator->get());
```

```
// получение новостей для текущей страницы
$rows = DB::query_range_fetch_all("SELECT * FROM {news}", $this->diafan->_paginator->polog, $this->diafan->_paginator->nastr);
```

Остальные свойства нужно добавлять по необходимости. Например, при поиске по товарам, чтобы добавить в навигацию условия поиска. Или для того, чтобы сделать дополнительную постраничную навигацию, как в комментариях.

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/js/paginator.show_more.js* – JS-сценарий модуля;
2. *modules/paginator/paginator.inc.php* – подключение для работы с постраничной навигацией;
3. *modules/paginator/paginator.install.php* – установка модуля;
4. *modules/paginator/views/paginator.view.get.php* – шаблон постраничной навигации для пользовательской части;
5. *modules/paginator/views/paginator.view.get_admin.php* – шаблон постраничной навигации для административной части;
6. *modules/paginator/views/paginator.view.show_more.php* – шаблон постраничной навигации для пользовательской части.

Карта сайта

- [Подключение](#)
- [База данных](#)
- [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 две части: страница сайта, к которой подключается модуль «Карта сайта» и файл для поисковых систем *sitemap.xml*.

На странице, выводимой пользователем выводятся все страницы сайта, а также категории подключенных модулей, согласно структуры сайта. Можно, также вывести элементы модуля (новости, товары, статьи и пр.). Код, выводящий элементы закомментирован в файле *modules/map/map.model.php*.

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'))
{
    $path = __FILE__; $i = 0;
    while(! file_exists($path.'/includes/404.php'))
    {
        if($i == 10) exit; $i++;
        $path = dirname($path);
    }
    include $path.'/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'),
    // 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;
}

```

Модуль индексирует ссылки, создаваемые модулями. Индекс обновляется в трех случаях:

- установка модуля «Карта сайта», сайт индексируется при первом открытии файла *sitemap.xml*
- установка модуля, индексируется весь устанавливаемый модуль
- редактирование элементов модуля, индексируются/удаляются из индекса отредактированные элементы

Для полной переиндексации сайта переустановите модуль «Карта сайта» в разделе «Модули и БД».

Согласно стандарту файл *sitemap.xml* может быть разбит на несколько файлов. Количество ссылок в одном файле задается свойством `private $max_url = 5000;` в файле *modules/map/map.sitemap.php*.

Подключение

Подключаемая часть – файл *modules/map/map.inc.php*. В нем описан класс *Map_inc*. В модуле к объекту класса можно обратиться через переменную `$this->diafan->_map`. Экземпляр класса создается при первом вызове переменной.

Методы

`void index_all ()` – Индексирует весь сайт.

Пример:

```

// индексирует все ссылки на сайте для файла sitemap.xml
$this->diafan->_map->index_all();

```

`void index_module (string $module_name)` – Индексирует модуль.

- string \$module_name: название модуля

Пример:


```
// индексирует модуль «Интернет магазин» для файла sitemap.xml
$this->diafan->_map->index_module('shop');
```

void **index_site_module** (array \$row, [boolean \$out = true]) – Индексирует страницы модуля, прикрепленного к странице сайта.

- array \$row: данные о странице сайта
- boolean \$out: функция используется как внешняя

Пример:

```
$row = array(
    "id" => 13,
    "module_name" => "shop",
    "timeedit" => 1398851484,
    "changefreq" => "always",
    "priority" => 0,
);
// индексирует страницу с прикрепленным модулем для файла sitemap.xml
$this->diafan->_map->index_site_module($row);
```

void **index_elements** (array \$rows) – Индексирует группу элементов.

- array \$rows: массив данных об индексируемых элементах

Пример:

```
$rows = array(
    array(
        "id" => 1,
        "element_type" => "element",
        "module_name" => "shop",
        "site_id" => 13,
        "timeedit" => 1398851484,
        "changefreq" => "always",
        "priority" => 0,
    ),
    array(
        "id" => 2,
        "element_type" => "element",
        "module_name" => "shop",
        "site_id" => 13,
        "timeedit" => 1398851485,
        "changefreq" => "always",
        "priority" => 0,
    ),
);
// индексирует несколько товаров для файла sitemap.xml
```

```
$this->diafan->_map->index_elements($rows);
```

void **index_element** (array \$row) – Индексирует один элемент.

- array \$row: данные об индексируемом элементе

Пример:

```
$row = array(  
    "id" => 1,  
    "element_type" => "element",  
    "module_name" => "shop",  
    "site_id" => 13,  
    "timeedit" => 1398851484,  
    "changefreq" => "always",  
    "priority" => 0,  
);  
// индексирует товар для файла sitemap.xml  
$this->diafan->_map->index_element($row);
```

void **delete** (integer|array \$element_ids, string \$module_name, [string \$element_type = 'element']) – Удаляет один или несколько элементов из индекса.

- integer|array \$element_ids: номер одного или нескольких элементов
- string \$module_name: название модуля
- string \$element_type: тип данных

Пример:

```
// удаляет производителя товаров из индекса для файла sitemap.xml  
$this->diafan->_map->delete(2, "shop", "brand");
```

void **delete_module** (string \$module_name) – Удаляет весь индекс модуля.

- string \$module_name: название модуля

Пример:

```
// удаляет все ссылки модуля «Новости» из файла sitemap.xml  
$this->diafan->_map->delete_module("news");
```

void **delete_sites** (array \$site_ids) – Удаляет индекс модулей, прикрепленных к страницам сайта.

- array \$site_ids: идентификаторы страниц сайта

Пример:

```
// удаляет все ссылки модулей, прикрепленных к страницам сайта ID 3, 4, 6 из файла  
sitemap.xml  
$this->diafan->_map->delete_sites(array(3, 4, 6));
```

База данных

[{map_index}](#) – Индекс для файла sitemap.xml

Файлы

1. *modules/map/admin/map.admin.inc.php* – подключение модуля к административной части других модулей;
2. *modules/map/js/map.tiny.js* – подгружает карту сайта в визуальном редакторе, JS-сценарий;
3. *modules/map/map.php* – контроллер;
4. *modules/map/map.inc.php* – подключение модуля;
5. *modules/map/map.install.php* – установка модуля;
6. *modules/map/map.model.php* – модель;
7. *modules/map/map.sitemap.php* – карта сайта в XML формате;
8. *modules/map/map.tiny.php* – плагин карты сайта для визуального редактора;
9. *modules/map/views/map.view.list.php* – шаблон списка страниц в карте сайте.

Captcha

- [Подключение](#)
- [Вопрос-Ответ](#)
- [Настройки](#)
- [База данных](#)
- [Файлы](#)

CAPTCHA – 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.

Примечание: в DIAFAN.CMS используется «reCAPTCHA» 2-ой версии.

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"], [string \$error = ""], [boolean \$is_update = false]) – Выводит капчу.

- string \$modules: метка капчи
- string \$error: ошибка ввода кода, если запрос передан не через Ajax
- boolean \$is_update: капча генерируется для обновления

Пример:

```
// в модели modules/comments/comments.inc.php запрашиваем капчу
$captcha = $this->diafan->_captcha->get("comments", "");

//выводим защитный код в шаблоне modules/comments/views/comments.view.form.php
echo $captcha;
```

boolean **configmodules** (string \$module, [integer \$site_id = 0]) – Проверяет подключена ли капта в настройках модуля.

- string \$module: названием модуля
- integer \$site_id: страница сайта с подключенным модулем

string|boolean false **error** ([string \$modules = "modules"]) – Проверяет правильность ввода капчи.

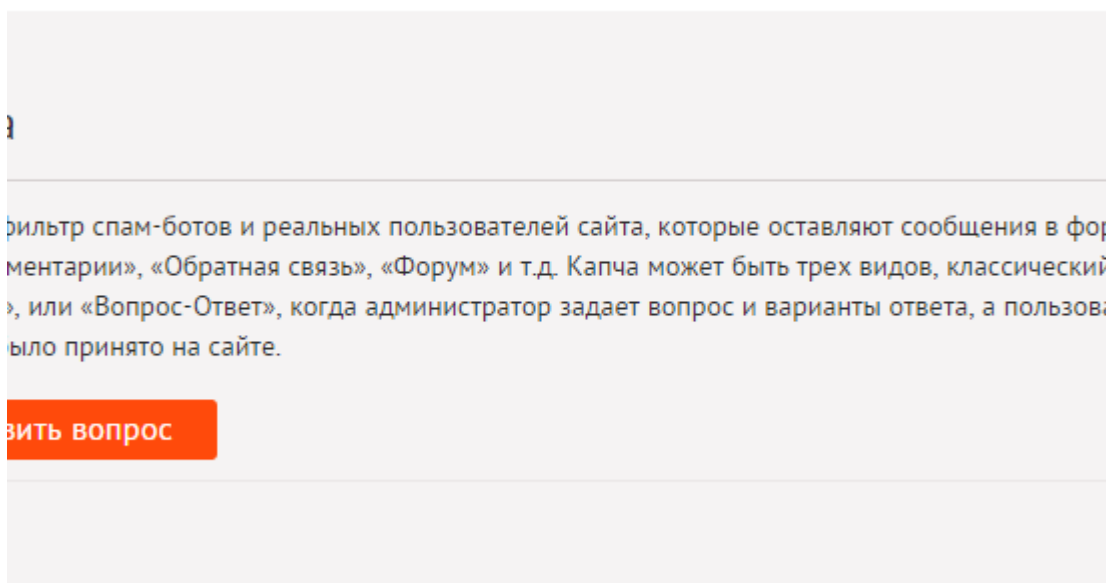
- string \$modules: метка капчи

Пример:

```
//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.action.php
$error = $this->diafan->_captcha->error('faq');
if($error)
{
    // защитный код введен неправильно
}
```

Вопрос-Ответ

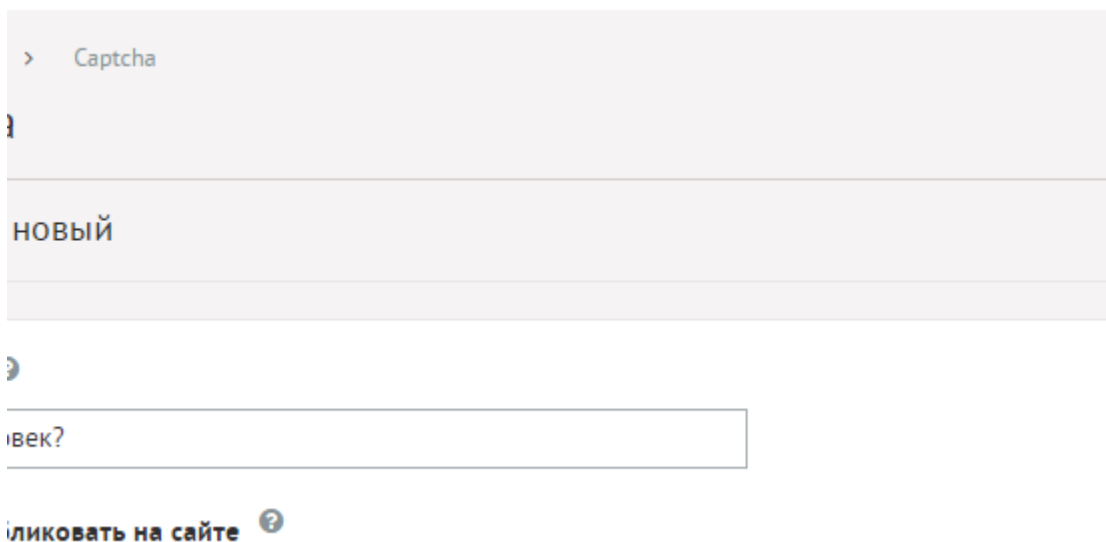


Фильтр спам-ботов и реальных пользователей сайта, которые оставляют сообщения в фор
ментарии», «Обратная связь», «Форум» и т.д. Капча может быть трех видов, классический
, или «Вопрос-Ответ», когда администратор задает вопрос и варианты ответа, а пользо
ыло принято на сайте.

Задать вопрос

НОВЫЙ

Бэкенд «Вопрос-Ответ» имеет свой административный интерфейс для ввода вопросов. Для добавления вопроса нужно заполнить следующие поля.



> Captcha

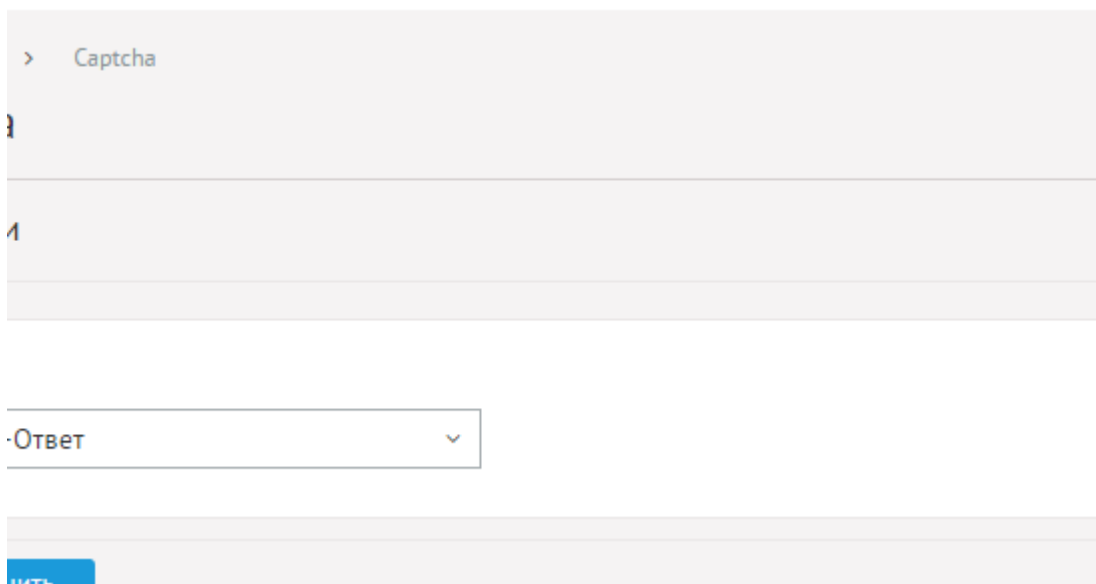
НОВЫЙ

Кликнуть на сайте ?

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.
- **Не показывать ответы** – Если отмечено, то пользователю на сайте будет предложено поле для ввода правильного ответа.
- **Ответы** – Возможные варианты ответов с указанием правильного.

Настройки



В настройках модуля можно выбрать бэкенд для капчи и вписать настройки для бэкенда (используется reCAPTCHA v.2).

- **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».

База данных

[{captcha}](#) – Вопросы для капчи типа Вопрос-ответ

[{captcha_answers}](#) – Варианты ответов для капчи типа Вопрос-ответ

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.inc.php* – подключение модуля к административной части других модулей;
4. *modules/captcha/admin/js/captcha.admin.config.js* – настройка модуля, JS-сценарий;
5. *modules/captcha/admin/js/captcha.admin.js* – редактирование вопросов для капчи, JS-сценарий;
6. *modules/captcha/captcha.get.php* – генерирование изображения капчи;
7. *modules/captcha/captcha.inc.php* – подключение для работы с капчей;
8. *modules/captcha/captcha.install.php* – установка модуля;
9. *modules/captcha/js/captcha.get.js* – JS-сценарий стандартной капчи;
10. *modules/captcha/js/captcha.get_qa.js* – JS-сценарий капчи «Вопрос-Ответ»;
11. *modules/captcha/js/captcha.get_recaptcha.js* – JS-сценарий для reCAPTCHA;
12. *modules/captcha/views/captcha.view.get.php* – шаблон стандартной капчи;
13. *modules/captcha/views/captcha.view.get_form.php* – шаблон формы стандартной капчи;
14. *modules/captcha/views/captcha.view.get_qa.php* – шаблон капчи «Вопрос-Ответ»;
15. *modules/captcha/views/captcha.view.get_qa_form.php* – шаблон формы для капчи «Вопрос-Ответ»;
16. *modules/captcha/views/captcha.view.get_recaptcha.php* – шаблон reCAPTCHA.

bbCode

- [Подключение](#)
- [Файлы](#)

[bbCode](#) — язык разметки, используемый для форматирования сообщений на многих электронных досках объявлений (BBS) и форумах.

Модуль позволяет подключить bbCode к сообщениям, добавляемых пользователями на сайте. В DIAFAN.CMS модуль подключен к модулям «Комментарии», «Форума» и «Личные сообщения». В настройках модулей bbCode можно подключить опцией «Использовать bbCode».

Подключение

Подключаемая часть – файл *modules/bbcode/bbcode.inc.php*. В нем описан класс *Bbcode_inc*. В модуле к объекту класса можно обратиться через переменную `$this->diafan->_bbcode`. Экземпляр класса создается при первом вызове переменной.

Методы

string **add (string \$text)** – Заменяет HTML-теги на bbCode.

- string \$text: текст

Пример:

```
// подготавливаем текст сообщения для редактирования
$text = $this->diafan->_bbcode->add($text);

echo '<textarea>'.$text.'</textarea>';
```

string **replace (string \$text, [boolean \$auto_url = true])** – Заменяет bbCode на HTML-теги.

- string \$text: текст
- boolean \$auto_url: автоопределение ссылок

Пример:

```
//обработываем текст сообщения перед добавлением в базу данных
$message = $this->diafan->_bbcode->replace($_POST["message"]);
```

В шаблоне модулей поле, в котором будет использоваться bbCode, нужно подключать следующим образом.

Пример:

```
// добавление bbCode к полю для ввода комментария
echo $this->get('get', 'bbcode', array("name" => "comment", "tag" => "username", "value" => ""));
```


Файлы

1. *modules/bbcode/bbcode.inc.php* – подключение для работы с bbCode;
2. *modules/bbcode/bbcode.install.php* – установка модуля;
3. *modules/bbcode/img/** – изображения;
4. *modules/bbcode/js/bbcode.get.js* – JS-обработка для поля сообщения;
5. *modules/bbcode/views/bbcode.view.get.php* – шаблон поля, для ввода сообщения.

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. Поэтому если Вам требуется сделать какое-то действие или доработку на сайте, скорее всего её можно сделать парой команд из API, чем переделкой ядра.
- 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* (за исключением кастомных тем, см.ниже).
- [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* (за исключением кастомных тем, см.ниже).
- Помните про [устройство шаблонов](#), и [кастомные темы сайта](#).

Архитектура DIAFAN.CMS

- [Общие данные](#)
- [Общие функции](#)
- [Объекты классов-подключений](#)

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 модуля;
6. подключение шаблона дизайна сайта (по умолчанию *themes/site.php*).

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. общие данные;
2. общие функции;
3. объекты классов-подключений.

Теперь подробнее об этих данных:

Общие данные

- `$this->diafan->installed_modules` – массив данных с именами модулей, которые подключены к фронтенду сайта (в таблице [modules](#) сделала запись о модуле `name=модуль, site='1'`);

Пример:

```
if(in_array('cart', $this->diafan->installed_modules))
{
    // модуль «Корзина товаров» установлен и можно к нему обращаться
```

```
}
```

- `$this->diafan->rewrite_variable_names` – список всех переменных, доступных для использования в URL;
- `$this->diafan->current_module` – модуль текущего исполняемого файла;
- `$this->diafan->current_insert_tag` – текущий шаблонный тег.

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();
```

Пример:

```
// редирект на главную страницу  
$this->diafan->redirect('/');
```

```
// получение сообщения для пользователя, заданного в настройках модуля «Обратная  
связь»  
$message = $this->diafan->configmodules('add_message', 'feedback');
```

Объекты классов-подключений

В основном объекте иницируются классы:

- кеширование (класс *Cache*, файл *includes/cache.php*, вызывается `$this->diafan->_cache`) – вся работа с кэшем;
- маршрутизатор (класс *Route*, файл *includes/route.php*, вызывается `$this->diafan->_route`) – доступ к переменным, передаваемым в URL, а также генерирование, сохранение, получение ЧПУ;
- шаблоны модулей (класс *Template*, файл *includes/template.php*, вызывается `$this->diafan->_tpl`) – подключение шаблонов модулей, обработка текста из визуального редактора;
- парсер шаблонных тегов (класс *Parser_theme*, файл *includes/parser_theme.php*, вызывается `$this->diafan->_parser_theme`) – подключает шаблонные теги, прописанные в виде тегов в шаблонах сайта и непосредственно в тексте визуального редактора.

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`, создаётся экземпляр класса.

Выделим отдельно подключения модулей, относящихся к ядру системы:

- страницы сайта – используется для получения данных о текущей странице и переопределение их в модуле.

Пример:

- `$this->diafan->_site->id` – номер текущей страницы;
 - `$this->diafan->_site->module` – модуль, прикрепленный к текущей странице;
 - `$this->diafan->_site->text` – контент текущей страницы.
- пользователи сайта – используется для получения сведений о текущем пользователе.

Пример:

- `$this->diafan->_users->id` – id-номер текущего пользователя;
 - `$this->diafan->_users->name` – логин текущего пользователя;
 - `$this->diafan->_users->fio` – ФИО текущего пользователя.
- языки сайта – используется для получения сведений о языках сайта, главной языке административной и пользовательской частей сайта и получение перевода интерфейса.

Пример:

- `$this->diafan->_languages->all` – массив данных обо всех языках, использующихся на сайте;
- `$this->diafan->_languages->site` – номер языка, использующегося как основной на сайте;
- `$this->diafan->_languages->admin` – номер языка, использующегося как основной в административной части.

Структура базы данных DIAFAN.CMS

DIAFAN.CMS состоит из следующих таблиц базы данных.

{ab}

Объявления

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
name	TEXT	название
act	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
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 для sitemap.xml
priority	VARCHAR(3)	Priority для sitemap.xml
noindex	ENUM('0', '1')	не индексировать: 0 - нет, 1 - да
cat_id	INT(11)	идентификатор основной категории из таблицы {ab_category}
site_id	INT(11)	идентификатор страницы сайта из таблицы {site}
keywords	VARCHAR(250)	ключевые слова, тег Keywords
descr	TEXT	описание, тэг Description
canonical	VARCHAR(100)	канонический тег
title_meta	VARCHAR(250)	заголовок окна в браузере, тег Title
anons	TEXT	анонс
anons_plus	ENUM('0', '1')	добавлять анонс к описанию: 0 - нет, 1 - да
text	TEXT	описание
prior	ENUM('0', '1')	важно, всегда сверху: 0 - нет, 1 - да
timeedit	INT(10)	время последнего изменения в формате UNIXTIME
access	ENUM('0', '1')	доступ ограничен: 0 - нет, 1 - да
admin_id	INT(11)	пользователь из таблицы {users} , добавивший или первый отредактировавший объявление в административной части
user_id	INT(11)	идентификатор пользователя из таблицы {users}
theme	VARCHAR(50)	шаблон страницы сайта
view	VARCHAR(50)	шаблон модуля
readed	ENUM('0', '1')	сообщение прочитано: 0 - нет, 1 - да
trash	ENUM('0', '1')	запись удалена в корзину: 0 - нет, 1 - да

{ab_category}

Категории объявлений

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
name	TEXT	название
act	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 для sitemap.xml
priority	VARCHAR(3)	Priority для sitemap.xml
noindex	ENUM('0', '1')	не индексировать: 0 - нет, 1 - да
parent_id	INT(11)	идентификатор родителя из таблицы {ab_category}
count_children	SMALLINT(5)	количество вложенных категорий
site_id	INT(11)	идентификатор страницы сайта из таблицы {site}
keywords	VARCHAR(250)	ключевые слова, тег Keywords
descr	TEXT	описание, тэг Description
canonical	VARCHAR(100)	канонический тег
title_meta	VARCHAR(250)	заголовок окна в браузере, тег Title
anons	TEXT	анонс
anons_plus	ENUM('0', '1')	добавлять анонс к описанию: 0 - нет, 1 - да
text	TEXT	описание
sort	INT(11)	порядковый номер для сортировки
timeedit	INT(10)	время последнего изменения в формате UNIXTIME
access	ENUM('0', '1')	доступ ограничен: 0 - нет, 1 - да
admin_id	INT(11)	пользователь из таблицы {users} , добавивший или первый отредктировавший категорию в административной части
theme	VARCHAR(50)	шаблон страницы сайта
view	VARCHAR(50)	шаблон модуля
view_rows	VARCHAR(50)	шаблон модуля для элементов в списке категории
view_element	VARCHAR(50)	шаблон модуля для элементов в категории
trash	ENUM('0', '1')	запись удалена в корзину: 0 - нет, 1 - да

{ab_category_parents}

Родительские связи категорий объявлений

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
element_id	INT(11)	идентификатор категории из таблицы

parent_id	INT(11)
trash	ENUM('0', '1')

[{ab_category}](#)

идентификатор категории-родителя из таблицы [{ab_category}](#)
запись удалена в корзину: 0 - нет, 1 - да

{ab_category_rel}

Связи объявлений и категорий

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
element_id	INT(11)	идентификатор объявления из таблицы <u>{ab}</u>
cat_id	INT(11)	идентификатор категории из таблицы <u>{ab_category}</u>
trash	ENUM('0', '1')	запись удалена в корзину: 0 - нет, 1 - да

{ab_counter}

Счетчик просмотров объявлений

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
element_id	INT(11)	идентификатор объявления из таблицы <u>{ab}</u>
count_view	INT(11)	количество просмотров
trash	ENUM('0', '1')	запись удалена в корзину: 0 - нет, 1 - да

{ab_param}

Дополнительные характеристики объявлений

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
name	VARCHAR(250)	название
type	VARCHAR(20)	тип
sort	INT(11)	порядковый номер для сортировки
search	ENUM('0', '1')	характеристика выводится в форме поиска: 0 - нет, 1 - да
list	ENUM('0', '1')	характеристика выводится в списке: 0 - нет, 1 - да
block	ENUM('0', '1')	характеристика выводится в блоке объявлений: 0 - нет, 1 - да
id_page	ENUM('0', '1')	характеристика выводится на странице объявления: 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
text	TEXT	описание
config	TEXT	дополнительные настройки поля
display_in_sort	ENUM('0', '1')	выводится в блоке для сортировки: 0 - нет, 1 - да

measure_unit	VARCHAR(50)	единица измерения
site_id	INT(11)	идентификатор страницы сайта из таблицы {site}
trash	ENUM('0', '1')	запись удалена в корзину: 0 - нет, 1 - да

{ab_param_category_rel}

Связи дополнительных характеристик объявлений и категорий

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
element_id	INT(11)	идентификатор характеристики из таблицы {ab_param}
cat_id	INT(11)	идентификатор категории из таблицы {ab_category}
trash	ENUM('0', '1')	запись удалена в корзину: 0 - нет, 1 - да

{ab_param_element}

Значений дополнительных характеристик объявлений

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
value	TEXT	значение
param_id	INT(11)	идентификатор характеристики из таблицы {ab_param}
element_id	INT(11)	идентификатор объявления из таблицы {ab}
trash	ENUM('0', '1')	запись удалена в корзину: 0 - нет, 1 - да

{ab_param_select}

Варианты значений дополнительных характеристик объявлений типа список

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
param_id	INT(11)	идентификатор характеристики из таблицы {ab_param}
value	TINYINT(2)	значение для типа характеристики «галочка»: 0 - нет, 1 - да
name	VARCHAR(50)	значение
sort	INT(11)	порядковый номер для сортировки
trash	ENUM('0', '1')	запись удалена в корзину: 0 - нет, 1 - да

{ab_rel}

Связи похожих объявлений

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
element_id	INT(11)	идентификатор объявления из таблицы {ab}
rel_element_id	INT(11)	идентификатор похожего объявления из таблицы {ab}

trash ENUM('0', '1')

запись удалена в корзину: 0 - нет, 1 - да

{access}

Доступ к элементам модулей

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
element_id	INT(11)	идентификатор элемента модуля
element_type	VARCHAR(20)	тип элемента модуля
module_name	VARCHAR(50)	название модуля
role_id	TINYINT(3)	идентификатор типа пользователя из таблицы {users_role}
trash	ENUM('0', '1')	запись удалена в корзину: 0 - нет, 1 - да

{addons}

Дополнения для CMS

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
addon_id	INT(11)	идентификатор дополнения
custom_id	INT(11)	идентификатор из таблицы {custom}
name	VARCHAR(100)	название
anons	TEXT	анонс
text	TEXT	описание
install	TEXT	описание установки дополнения
link	VARCHAR(255)	внешняя ссылка
image	VARCHAR(255)	внешняя ссылка на изображение
author	TEXT	данные об авторе
author_link	TEXT	ссылка на страницу автора
sort	INT(11)	порядковый номер для сортировки
downloaded	INT(11)	количество скачиваний
timeedit	INT(10)	время последнего изменения в формате UNIXTIME
custom_timeedit	INT(10)	время последнего изменения в формате UNIXTIME в таблице {custom}
import_update	ENUM('0', '1')	метка обновления записи: 0 - нет, 1 - да

{admin}

Страницы в админки

Поле	Тип (размерность)	Описание
id	SMALLINT(5)	идентификатор
parent_id	SMALLINT(5)	идентификатор родителя из таблицы {admin}
count_children	SMALLINT(5)	количество вложенных страниц
group_id	ENUM('1', '2', '3', '4', '5')	группа
name	VARCHAR(100)	название
rewrite	VARCHAR(30)	псевдоссылка

act	ENUM('0', '1')	показывать в меню: 0 - нет, 1 - да
add	ENUM('0', '1')	ссылка на добавление элемента в быстром меню: 0 - нет, 1 - да
add_name	VARCHAR(100)	текст ссылки на добавление элемента в быстром меню
sort	SMALLINT(5)	порядковый номер для сортировки
docs	VARCHAR(255)	ссылка на раздел в документации

{admin_parents}

Родительские связи страниц админки

Поле	Тип (размерность)	Описание
id	SMALLINT(5)	идентификатор
element_id	SMALLINT(5)	идентификатор страницы из таблицы {admin}
parent_id	SMALLINT(5)	идентификатор страницы родителя из таблицы {admin}

{attachments}

Прикрепленные файлы

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
name	VARCHAR(50)	название файла
module_name	VARCHAR(50)	название модуля
element_id	INT(11)	идентификатор элемента модуля
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

{balance}

Баланс пользователей

Поле	Тип (размерность)	Описание
id	SMALLINT(5)	идентификатор
user_id	INT(11)	идентификатор пользователя из таблицы {users}
summ	DOUBLE	сумма на балансе

{bs}

Баннеры

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор

name	VARCHAR(250)	название
text	TEXT	описание
alt	VARCHAR(250)	атрибут alt для баннера-изображения
title	VARCHAR(250)	атрибут title для баннера-изображения
act	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)	имя файла, загруженного в папку userfls/bs
html	TEXT	HTML код баннера-блока
link	TEXT	ссылка на баннер-изображение или баннер-флэш
cat_id	INT(11)	идентификатор категории из таблицы {bs_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
created	INT(10)	дата создания
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
sort	INT(11)	порядковый номер для сортировки
trash	ENUM('0', '1')	запись удалена в корзину: 0 - нет, 1 - да

{bs_category}

Категории баннеров

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
name	TEXT	название
act	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

{bs_site_rel}

Данные о том, на каких страницах сайта выводятся баннеры

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
element_id	INT(11)	идентификатор баннера из таблицы {bs_category}
site_id	INT(11)	идентификатор страницы сайта из таблицы {site}
trash	ENUM('0', '1')	запись удалена в корзину: 0 - нет, 1 - да

{captcha}

Вопросы для капчи типа Вопрос-ответ

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
name	TEXT	текст вопроса
act	ENUM('0', '1')	показывать на сайте: 0 - нет, 1 - да
is_write	ENUM('0', '1')	ответы не отображаются на сайте: 0 - нет, 1 - да
trash	ENUM('0', '1')	запись удалена в корзину: 0 - нет, 1 - да

{captcha_answers}

Варианты ответов для капчи типа Вопрос-ответ

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
text	TEXT	текст ответа
is_right	ENUM('0', '1')	ответ правильный: 0 - нет, 1 - да
captcha_id	INT(11)	идентификатор вопроса из таблицы {captcha}
trash	ENUM('0', '1')	запись удалена в корзину: 0 - нет, 1 - да

{clauses}

Статьи

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
name	TEXT	название
act	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 для sitemap.xml
priority	VARCHAR(3)	Priority для sitemap.xml
noindex	ENUM('0', '1')	не индексировать: 0 - нет, 1 - да
created	INT(10)	дата создания
date_start	INT(10)	дата начала показа
date_finish	INT(10)	дата окончания показа
cat_id	INT(11)	идентификатор основной категории из таблицы {clauses_category}

site_id	INT(11)	идентификатор страницы сайта из таблицы {site}
keywords	VARCHAR(250)	ключевые слова, тег Keywords
descr	TEXT	описание, тэг Description
canonical	VARCHAR(100)	канонический тег
title_meta	VARCHAR(250)	заголовок окна в браузере, тег Title
anons	TEXT	анонс
anons_plus	ENUM('0', '1')	добавлять анонс к описанию: 0 - нет, 1 - да
text	LONGTEXT	описание
timeedit	INT(10)	время последнего изменения в формате UNIXTIME
admin_id	INT(11)	пользователь из таблицы {users} , добавивший или первый отредактировавший статью в административной части
access	ENUM('0', '1')	доступ ограничен: 0 - нет, 1 - да
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}

Категории статей

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
name	TEXT	название
act	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 для sitemap.xml
priority	VARCHAR(3)	Priority для sitemap.xml
noindex	ENUM('0', '1')	не индексировать: 0 - нет, 1 - да
parent_id	INT(11)	идентификатор родителя из таблицы {clauses_category}
count_children	SMALLINT(5)	количество вложенных категорий
site_id	INT(11)	идентификатор страницы сайта из таблицы {site}
keywords	VARCHAR(250)	ключевые слова, тег Keywords
descr	TEXT	описание, тэг Description
canonical	VARCHAR(100)	канонический тег
title_meta	VARCHAR(250)	заголовок окна в браузере, тег Title
anons	TEXT	анонс
anons_plus	ENUM('0', '1')	добавлять анонс к описанию: 0 - нет, 1 - да
text	TEXT	описание

sort	INT(11)	порядковый номер для сортировки
timeedit	INT(10)	время последнего изменения в формате UNIXTIME
admin_id	INT(11)	пользователь из таблицы {users} , добавивший или первый отредктировавший категорию в административной части
access	ENUM('0', '1')	доступ ограничен: 0 - нет, 1 - да
theme	VARCHAR(50)	шаблон страницы сайта
view	VARCHAR(50)	шаблон модуля
view_rows	VARCHAR(50)	шаблон модуля для элементов в списке категории
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}

Родительские связи категорий статей

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
element_id	INT(11)	идентификатор категории из таблицы {clauses_category}
parent_id	INT(11)	идентификатор категории-родителя из таблицы {clauses_category}
trash	ENUM('0', '1')	запись удалена в корзину: 0 - нет, 1 - да

{clauses_category_rel}

Связи статей и категорий

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
element_id	INT(11)	идентификатор статьи из таблицы {clauses}
cat_id	INT(11)	идентификатор категории из таблицы {clauses_category}
trash	ENUM('0', '1')	запись удалена в корзину: 0 - нет, 1 - да

{clauses_counter}

Счетчик просмотров статей

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
element_id	INT(11)	идентификатор статьи из таблицы {clauses}
count_view	INT(11)	количество просмотров
trash	ENUM('0', '1')	запись удалена в корзину: 0 - нет, 1 - да

{clauses_rel}

Связи похожих статей

Поле	Тип (размерность)	Описание
------	-------------------	----------

id	INT(11)	идентификатор
element_id	INT(11)	идентификатор статьи из таблицы {clauses}
rel_element_id	INT(11)	идентификатор связанной статьи из таблицы {clauses}
trash	ENUM('0', '1')	запись удалена в корзину: 0 - нет, 1 - да

{comments}

Комментарии

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
user_id	INT(11)	идентификатор пользователя из таблицы {users}
text	TEXT	текст комментария
act	ENUM('0', '1')	показывать на сайте: 0 - нет, 1 - да
created	INT(10)	дата создания
module_name	VARCHAR(50)	название модуля
element_type	ENUM('element', 'cat')	тип элемента модуля
element_id	INT(11)	идентификатор элемента модуля
parent_id	INT(11)	идентификатор родителя из таблицы {comments}
count_children	SMALLINT(5)	количество вложенных комментариев
trash	ENUM('0', '1')	запись удалена в корзину: 0 - нет, 1 - да

{comments_mail}

Подписка пользователей на новые комментарии

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
mail	VARCHAR(64)	e-mail пользователя
module_name	VARCHAR(50)	название модуля
element_id	INT(11)	идентификатор элемента модуля
element_type	ENUM('element', 'cat')	тип элемента модуля
trash	ENUM('0', '1')	запись удалена в корзину: 0 - нет, 1 - да

{comments_param}

Дополнительные поля комментариев

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
name	VARCHAR(250)	название
type	VARCHAR(20)	тип
module_name	VARCHAR(50)	название модуля
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_a	ENUM('0', '1')	показывать в форме авторизованным

uth		пользователям: 0 - нет, 1 - да
show_in_form_n	ENUM('0', '1')	показывать в списке: 0 - нет, 1 - да
o_auth		
text	TEXT	описание
config	TEXT	сериализованные настройки поля
trash	ENUM('0', '1')	запись удалена в корзину: 0 - нет, 1 - да

{comments_param_element}

Значения дополнительных полей комментариев

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
value	TEXT	значение
param_id	INT(11)	идентификатор характеристики 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}

Варианты значения дополнительных полей комментариев с типом список

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
param_id	INT(11)	идентификатор характеристики из таблицы {comments_param}
name	VARCHAR(50)	значение
value	VARCHAR(1)	значение для типа характеристики «галочка»: 0 - нет, 1 - да
sort	INT(11)	порядковый номер для сортировки
trash	ENUM('0', '1')	запись удалена в корзину: 0 - нет, 1 - да

{comments_parents}

Родительские связи комментариев

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
element_id	INT(11)	идентификатор комментария из таблицы {comments}
parent_id	INT(11)	идентификатор родителя комментария из таблицы {comments}
trash	ENUM('0', '1')	запись удалена в корзину: 0 - нет, 1 - да

{config}

Настройки модулей

Поле	Тип (размерность)	Описание
id	SMALLINT(5)	идентификатор

module_name	VARCHAR(50)	название модуля
name	VARCHAR(50)	название настройки
lang_id	SMALLINT(5)	идентификатор языка сайта из таблицы {languages}
value	TEXT	значение
site_id	INT(11)	идентификатор страницы сайта из таблицы {site}

{custom}

Темы

Поле	Тип (размерность)	Описание
id	SMALLINT(5)	идентификатор
name	VARCHAR(100)	название
created	INT(10)	дата создания
current	ENUM('0', '1')	текущая тема: 0 - нет, 1 - да
text	TEXT	описание

{faq}

Вопросы и ответы

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
name	TEXT	название
act	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 для sitemap.xml
priority	VARCHAR(3)	Priority для sitemap.xml
noindex	ENUM('0','1')	не индексировать: 0 - нет, 1 - да
created	INT(10)	дата создания
date_start	INT(10)	дата начала показа
date_finish	INT(10)	дата окончания показа
mail	VARCHAR(40)	е-mail пользоваля, задавшего вопрос
cat_id	INT(11)	идентификатор основной категории из таблицы {faq_category}
site_id	INT(11)	идентификатор страницы сайта из таблицы {site}
keywords	VARCHAR(250)	ключевые слова, тег Keywords
descr	TEXT	описание, тэг Description
canonical	VARCHAR(100)	канонический тег
title_meta	VARCHAR(250)	заголовок окна в браузере, тег Title
anons	TEXT	вопрос
text	TEXT	ответ
often	ENUM('0', '1')	часто задаваемый вопрос: 0 - нет, 1 - да

timeedit	INT(10)	время последнего изменения в формате UNIXTIME
access	ENUM('0', '1')	доступ ограничен: 0 - нет, 1 - да
user_id	INT(11)	идентификатор пользователя из таблицы {users}
admin_id	INT(11)	пользователь из таблицы {users} , добавивший или первый отредктировавший вопрос в административной части
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}

Категории вопросов и ответов

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
name	TEXT	название
act	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 для sitemap.xml
priority	VARCHAR(3)	Priority для sitemap.xml
noindex	ENUM('0','1')	не индексировать: 0 - нет, 1 - да
parent_id	INT(11)	идентификатор родителя из таблицы {faq_category}
count_children	SMALLINT(5)	количество вложенных категорий
site_id	INT(11)	идентификатор страницы сайта из таблицы {site}
keywords	VARCHAR(250)	ключевые слова, тег Keywords
descr	TEXT	описание, тэг Description
canonical	VARCHAR(100)	канонический тег
title_meta	VARCHAR(250)	заголовок окна в браузере, тег Title
anons	TEXT	анонс
anons_plus	ENUM('0', '1')	добавлять анонс к описанию: 0 - нет, 1 - да
text	TEXT	описание
sort	INT(11)	порядковый номер для сортировки
timeedit	INT(10)	время последнего изменения в формате UNIXTIME
access	ENUM('0', '1')	доступ ограничен: 0 - нет, 1 - да
admin_id	INT(11)	пользователь из таблицы {users} , добавивший или первый отредктировавший категорию в административной части
theme	VARCHAR(50)	шаблон страницы сайта

view	VARCHAR(50)	шаблон модуля
view_rows	VARCHAR(50)	шаблон модуля для элементов в списке категории
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}

Родительские связи категорий вопросов и ответов

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}

Связи вопросов и ответов с категориями

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}

Счетчик вопросов и ответов

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
element_id	INT(11)	идентификатор вопроса из таблицы {faq}
count_view	INT(11)	количество просмотров
trash	ENUM('0', '1')	запись удалена в корзину: 0 - нет, 1 - да

{faq_rel}

Связи похожих вопросов и ответов

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}

Сообщения из обратной связи

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)	идентификатор языка сайта из таблицы {languages}
text	TEXT	ответ
user_id	INT(11)	идентификатор пользователя из таблицы {users}
admin_id	INT(11)	пользователь из таблицы {users} , добавивший или первый отредактировавший сообщение в административной части
readed	ENUM('0', '1')	сообщение прочитано: 0 - нет, 1 - да
url	TEXT	страница, с которой отправлено сообщение
trash	ENUM('0', '1')	запись удалена в корзину: 0 - нет, 1 - да

{feedback_param}

Поля конструктора формы обратной связи

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
name	VARCHAR(250)	название
type	VARCHAR(20)	тип
site_id	INT(11)	идентификатор страницы сайта из таблицы {site}
sort	INT(11)	порядковый номер для сортировки
required	ENUM('0', '1')	обязательно для заполнения: 0 - нет, 1 - да
text	TEXT	описание
config	TEXT	дополнительные настройки поля
trash	ENUM('0', '1')	запись удалена в корзину: 0 - нет, 1 - да

{feedback_param_element}

Значения полей конструктора формы обратной связи, заполненные в сообщении

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
value	TEXT	значение
param_id	INT(11)	идентификатор характеристики 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}

Варианты значений полей конструктора с типом список

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
param_id	INT(11)	идентификатор характеристики из таблицы {feedback_param}
value	TINYINT(2)	значение для типа характеристики «галочка»: 0 - нет, 1 - да
name	VARCHAR(50)	значение
sort	INT(11)	порядковый номер для сортировки
trash	ENUM('0', '1')	запись удалена в корзину: 0 - нет, 1 - да

{files}

Файлы в файловом архиве

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
name	TEXT	название
act	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 для sitemap.xml
priority	VARCHAR(3)	Priority для sitemap.xml
noindex	ENUM('0','1')	не индексировать: 0 - нет, 1 - да
cat_id	INT(11)	идентификатор основной категории из таблицы {files_category}
site_id	INT(11)	идентификатор страницы сайта из таблицы {site}
keywords	VARCHAR(250)	ключевые слова, тег Keywords
descr	TEXT	описание, тэг Description
canonical	VARCHAR(100)	канонический тег
title_meta	VARCHAR(250)	заголовок окна в браузере, тег Title
anons	TEXT	анонс
anons_plus	ENUM('0', '1')	добавлять анонс к описанию: 0 - нет, 1 - да
text	TEXT	описание
sort	INT(11)	порядковый номер для сортировки
timeedit	INT(10)	время последнего изменения в формате UNIXTIME
access	ENUM('0', '1')	доступ ограничен: 0 - нет, 1 - да
admin_id	INT(11)	пользователь из таблицы {users} , добавивший или первый отредактировавший файл в административной части
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}

Категории файлового архива

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
name	TEXT	название
act	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 для sitemap.xml
priority	VARCHAR(3)	Priority для sitemap.xml
noindex	ENUM('0', '1')	не индексировать: 0 - нет, 1 - да
parent_id	INT(11)	идентификатор родителя из таблицы {files_category}
count_children	SMALLINT(5)	количество вложенных категорий
site_id	INT(11)	идентификатор страницы сайта из таблицы {site}
keywords	VARCHAR(250)	ключевые слова, тег Keywords
descr	TEXT	описание, тэг Description
canonical	VARCHAR(100)	канонический тег
title_meta	VARCHAR(250)	заголовок окна в браузере, тег Title
anons	TEXT	анонс
anons_plus	ENUM('0', '1')	добавлять анонс к описанию: 0 - нет, 1 - да
text	TEXT	описание
sort	INT(11)	порядковый номер для сортировки
timeedit	INT(10)	время последнего изменения в формате UNIXTIME
access	ENUM('0', '1')	доступ ограничен: 0 - нет, 1 - да
admin_id	INT(11)	пользователь из таблицы {users} , добавивший или первый отредактировавший категорию в административной части
theme	VARCHAR(50)	шаблон страницы сайта
view	VARCHAR(50)	шаблон модуля
view_rows	VARCHAR(50)	шаблон модуля для элементов в списке категории
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}

Родительские связи категорий файлового архива

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}

Связи файлов и категорий в файловом архиве

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}

Счетчик просмотров файлов в файловом архиве

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
element_id	INT(11)	идентификатор файла из таблицы {files}
count_view	INT(11)	количество просмотров
trash	ENUM('0', '1')	запись удалена в корзину: 0 - нет, 1 - да

{files_links}

Внешние ссылки на файлы для файлового архива

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
element_id	INT(11)	идентификатор файла из таблицы {files}
link	VARCHAR(255)	внешняя ссылка
trash	ENUM('0', '1')	запись удалена в корзину: 0 - нет, 1 - да

{files_rel}

Связи похожих файлов

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
element_id	INT(11)	идентификатор вопроса из таблицы {files}
rel_element_id	INT(11)	идентификатор связанного вопроса из таблицы {files}
trash	ENUM('0', '1')	запись удалена в корзину: 0 - нет, 1 - да

{forum}

Темы форума

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор

user_id	INT(11)	идентификатор пользователя из таблицы {users}
cat_id	INT(11)	идентификатор категории из таблицы {forum_category}
created	INT(10)	дата создания
name	VARCHAR(250)	название
date_update	INT(10)	дата редакции
user_update	INT(11)	идентификатор пользователя из таблицы {users} , отредактировавший сообщение
timeedit	INT(11)	время последнего изменения в формате UNIXTIME
counter_view	INT(11)	количество просмотров
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_blocks}

Блоки форума

Поле	Тип (размерность)	Описание
id	SMALLINT(5)	идентификатор
name	VARCHAR(250)	название
sort	INT(11)	порядковый номер для сортировки
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}

Категории форума

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
name	VARCHAR(250)	название
timeedit	INT(10)	время последнего изменения в формате UNIXTIME
block_id	SMALLINT(5)	идентификатор блока из таблицы {forum_blocks}
counter_view	INT(11)	количество просмотров
act	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

{forum_messages}

Сообщения на форуме

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор

name	VARCHAR(20)	название
user_id	INT(11)	идентификатор пользователя из таблицы {users}
parent_id	INT(11)	идентификатор родителя из таблицы {forum_messages}
count_children	SMALLINT(5)	количество вложенных сообщений
created	INT(10)	дата создания
date_update	INT(10)	дата редакции
user_update	INT(11)	идентификатор пользователя из таблицы {users} , отредактировавший сообщение
forum_id	INT(11)	идентификатор темы из таблицы {forum}
text	TEXT	описание
act	ENUM('0', '1')	показывать на сайте: 0 - нет, 1 - да
trash	ENUM('0', '1')	запись удалена в корзину: 0 - нет, 1 - да

{forum_messages_parents}

Родительские связи сообщений

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
element_id	INT(11)	идентификатор сообщения из таблицы {forum_messages}
parent_id	INT(11)	идентификатор родителя сообщения из таблицы {forum_messages}
trash	ENUM('0', '1')	запись удалена в корзину: 0 - нет, 1 - да

{forum_show}

Новые сообщения и темы для пользователей

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
element_id	INT(11)	идентификатор сообщения из таблицы {forum_messages}
user_id	INT(11)	идентификатор пользователя из таблицы {users}
created	INT(10)	дата создания
trash	ENUM('0', '1')	запись удалена в корзину: 0 - нет, 1 - да

{geomap}

Точки на карте

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
element_id	INT(11)	идентификатор элемента модуля
module_name	VARCHAR(50)	название модуля
element_type	ENUM('element', 'cat')	тип элемента модуля
point	TEXT	координаты точки

trash	ENUM('0', '1')	запись удалена в корзину: 0 - нет, 1 - да
-------	----------------	---

{images}

Прикрепленные изображения

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
image_id	INT(11)	идентификатор изображения из таблицы {images} , если запись является ссылкой на изображение
name	VARCHAR(250)	название файла
module_name	VARCHAR(50)	название модуля
element_type	ENUM('element', 'cat', 'brand')	тип элемента модуля
element_id	INT(11)	идентификатор элемента модуля
alt	VARCHAR(250)	атрибут alt
title	VARCHAR(250)	атрибут title
param_id	INT(11)	идентификатор поля или характеристики с типом «изображения»
size	INT(11)	размер файла
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
folder_num	TINYINT(3)	номер папки
trash	ENUM('0', '1')	запись удалена в корзину: 0 - нет, 1 - да

{images_editor_folders}

Папки изображений в плагине для визуального редактора

Поле	Тип (размерность)	Описание
id	SMALLINT(5)	идентификатор
parent_id	SMALLINT(5)	идентификатор папки-родителя из таблицы {images_editor_folders}
name	VARCHAR(250)	название

{images_variations}

Варианты загрузки 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}

Ключевые слова для перелинковки

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
text	TEXT	ключевое слово
link	TEXT	ссылка
act	ENUM('0', '1')	показывать на сайте: 0 - нет, 1 - да
trash	ENUM('0', '1')	запись удалена в корзину: 0 - нет, 1 - да

{languages}

Языки сайта

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}

Перевод интерфейса

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
text	TEXT	описание
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}

Лог 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}

Лог голосований и оценок

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
module_name	VARCHAR(50)	название модуля
element_id	INT(11)	идентификатор элемента модуля
element_type	ENUM('element', 'cat')	тип элемента модуля
note	INT(7)	оценка
created	INT(10)	дата создания
ip	VARCHAR(62)	IP-адрес пользователя
session_id	VARCHAR(64)	идентификатор сессии из таблицы {sessions}
include_name	VARCHAR(10)	подключенный модуль

{map_index}

Индекс для файла sitemap.xml

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
url	TEXT	ссылка
module_name	VARCHAR(50)	название модуля
element_id	INT(11)	идентификатор элемента модуля
element_type	ENUM('element', 'cat', 'param', 'brand')	тип элемента модуля
timeedit	INT(10)	время последнего редактирования
changefreq	ENUM('always', 'hourly', 'daily', 'weekly', 'monthly', 'yearly', 'never')	Changefreq
priority	VARCHAR(3)	Priority
date_start	INT(10)	дата начала показа
date_finish	INT(10)	дата окончания показа
trash	ENUM('0', '1')	запись удалена в корзину: 0 - нет, 1 - да

{menu}

Пункты меню

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
name	VARCHAR(100)	название
module_name	VARCHAR(50)	название модуля
element_type	ENUM('element', 'cat', 'brand', 'param')	тип элемента модуля
element_id	INT(11)	идентификатор элемента модуля
cat_id	INT(11)	идентификатор меню из таблицы {menu_category}
parent_id	INT(11)	идентификатор пункта меню - родителя из таблицы {menu}
count_children	SMALLINT(5)	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
act	ENUM('0', '1')	показывать на сайте: 0 - нет, 1 - да
text	VARCHAR(255)	описание пункта меню
attributes	VARCHAR(250)	атрибуты ссылки
target_blank	ENUM('0', '1')	открывать в новом окне
trash	ENUM('0', '1')	запись удалена в корзину: 0 - нет, 1 - да

{menu_category}

Меню

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
name	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
act	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
sort	INT(11)	порядковый номер для сортировки
trash	ENUM('0', '1')	запись удалена в корзину: 0 - нет, 1 - да

{menu_category_site_rel}

Данные о том, на каких страницах сайта выводится меню

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
element_id	INT(11)	идентификатор пункта меню из таблицы {menu}
site_id	INT(11)	идентификатор страницы сайта из таблицы {site}
trash	ENUM('0', '1')	запись удалена в корзину: 0 - нет, 1 - да

{menu_parents}

Родительские связи пунктов меню

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
element_id	INT(11)	идентификатор пункта меню из таблицы

parent_id	INT(11)
trash	ENUM('0', '1')

{menu}

идентификатор пункта меню - родителя из таблицы {menu}
запись удалена в корзину: 0 - нет, 1 - да

{messages}

Личные сообщения пользователей

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
author	INT(11)	идентификатор пользователя из таблицы <u>{users}</u> , написавшего сообщение
to_user	INT(11)	идентификатор пользователя из таблицы <u>{users}</u> получателя сообщения
text	TEXT	текст сообщения
created	INT(10)	дата создания
readed	ENUM('0', '1')	сообщение прочитано: 0 - нет, 1 - да
trash	ENUM('0', '1')	запись удалена в корзину: 0 - нет, 1 - да

{messages_user}

Контакты пользователей

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
user_id	INT(11)	идентификатор пользователя из таблицы <u>{users}</u> кто инициировал контакт
contact_user_id	INT(11)	идентификатор пользователя из таблицы <u>{users}</u> с кем создан контакт
date_update	INT(10)	дата последнего сообщения
readed	ENUM('0', '1')	все сообщения прочитаны: 0 - нет, 1 - да
count_message	SMALLINT(5)	количество сообщений
trash	ENUM('0', '1')	запись удалена в корзину: 0 - нет, 1 - да

{mistakes}

Ошибки на сайте, добавленные пользователями

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}

Модули

Поле	Тип (размерность)	Описание
id	SMALLINT(5)	идентификатор

name	VARCHAR(30)	название
module_name	VARCHAR(50)	название основного модуля
site	ENUM('0', '1')	используется на сайте: 0 - нет, 1 - да
site_page	ENUM('0', '1')	подключается к странице сайта: 0 - нет, 1 - да
admin	ENUM('0', '1')	используется в административной части: 0 - нет, 1 - да
title	VARCHAR(100)	название для пользователей

{news}

Новости

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
name	TEXT	название
act	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 для sitemap.xml
priority	VARCHAR(3)	Priority для sitemap.xml
noindex	ENUM('0','1')	не индексировать: 0 - нет, 1 - да
created	INT(10)	дата создания
date_start	INT(10)	дата начала показа
date_finish	INT(10)	дата окончания показа
cat_id	INT(11)	идентификатор основной категории из таблицы {news_category}
site_id	INT(11)	идентификатор страницы сайта из таблицы {site}
keywords	VARCHAR(250)	ключевые слова, тег Keywords
descr	TEXT	описание, тэг Description
canonical	VARCHAR(100)	канонический тег
title_meta	VARCHAR(250)	заголовок окна в браузере, тег Title
anons	TEXT	анонс
anons_plus	ENUM('0', '1')	добавлять анонс к описанию: 0 - нет, 1 - да
text	TEXT	описание
prior	ENUM('0', '1')	важно, всегда сверху: 0 - нет, 1 - да
timeedit	INT(10)	время последнего изменения в формате UNIXTIME
admin_id	INT(11)	пользователь из таблицы {users} , добавивший или первый отредактировавший новость в административной части
access	ENUM('0', '1')	доступ ограничен: 0 - нет, 1 - да
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}

Категории новостей

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
name	TEXT	название
act	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 для sitemap.xml
priority	VARCHAR(3)	Priority для sitemap.xml
noindex	ENUM('0', '1')	не индексировать: 0 - нет, 1 - да
parent_id	INT(11)	идентификатор родителя из таблицы {news_category}
count_children	SMALLINT(5)	количество вложенных категорий
site_id	INT(11)	идентификатор страницы сайта из таблицы {site}
keywords	VARCHAR(250)	ключевые слова, тег Keywords
descr	TEXT	описание, тэг Description
canonical	VARCHAR(100)	канонический тег
title_meta	VARCHAR(250)	заголовок окна в браузере, тег Title
anons	TEXT	анонс
anons_plus	ENUM('0', '1')	добавлять анонс к описанию: 0 - нет, 1 - да
text	TEXT	описание
sort	INT(11)	порядковый номер для сортировки
timeedit	INT(10)	время последнего изменения в формате UNIXTIME
admin_id	INT(11)	пользователь из таблицы {users} , добавивший или первый отредактировавший категорию в административной части
access	ENUM('0', '1')	доступ ограничен: 0 - нет, 1 - да
theme	VARCHAR(50)	шаблон страницы сайта
view	VARCHAR(50)	шаблон модуля
view_rows	VARCHAR(50)	шаблон модуля для элементов в списке категории
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}

Родительские связи категорий новостей

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
element_id	INT(11)	идентификатор категории из таблицы

parent_id	INT(11)
trash	ENUM('0', '1')

[{news_category}](#)

идентификатор категории-родителя из таблицы [{news_category}](#)

запись удалена в корзину: 0 - нет, 1 - да

{news_category_rel}

Связи новостей и категорий

Поле	Тип (размерность)
id	INT(11)
element_id	INT(11)
cat_id	INT(11)
trash	ENUM('0', '1')

Описание

идентификатор

идентификатор новости из таблицы [{news}](#)

идентификатор категории из таблицы [{news_category}](#)

запись удалена в корзину: 0 - нет, 1 - да

{news_counter}

Счетчик просмотров новостей

Поле	Тип (размерность)
id	INT(11)
element_id	INT(11)
count_view	INT(11)
trash	ENUM('0', '1')

Описание

идентификатор

идентификатор новости из таблицы [{news}](#)

количество просмотров

запись удалена в корзину: 0 - нет, 1 - да

{news_rel}

Связи похожих новостей

Поле	Тип (размерность)
id	INT(11)
element_id	INT(11)
rel_element_id	INT(11)
trash	ENUM('0', '1')

Описание

идентификатор

идентификатор новости из таблицы [{news}](#)

идентификатор связанной новости из таблицы [{news}](#)

запись удалена в корзину: 0 - нет, 1 - да

{payment}

Методы оплаты

Поле	Тип (размерность)
id	SMALLINT(5)
name	VARCHAR(50)
text	TEXT
act	ENUM('0', '1')
payment	VARCHAR(20)
params	TEXT
trash	ENUM('0', '1')
sort	INT(11)

Описание

идентификатор

название

описание

показывать на сайте: 0 - нет, 1 - да

платежная система

сериализованные настройки платежной системы

запись удалена в корзину: 0 - нет, 1 - да

порядковый номер для сортировки

{payment_history}

История платежей

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
element_id	INT(11)	элемент модуля, для которого производится платеж (номер заказа или идентификатор пользователя)
status	ENUM('request_pay', 'pay')	статус платежа: request_pay - запрошен, pay - оплачен
created	INT(10)	дата создания
summ	DOUBLE	сумма платежа
payment_id	INT(10)	идентификатор метода оплаты из таблицы {payment}
module_name	ENUM('cart', 'balance')	модуль, в котором используются платежи
code	VARCHAR(32)	код доступа к платежным документам
payment_data	VARCHAR(50)	данные о платеже из платежной системы

{photo}

Фотографии

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
name	TEXT	название
act	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 для sitemap.xml
priority	VARCHAR(3)	Priority для sitemap.xml
noindex	ENUM('0','1')	не индексировать: 0 - нет, 1 - да
cat_id	INT(11)	идентификатор основного альбома из таблицы {photo_category}
site_id	INT(11)	идентификатор страницы сайта из таблицы {site}
keywords	VARCHAR(250)	ключевые слова, тег Keywords
descr	TEXT	описание, тэг Description
canonical	VARCHAR(100)	канонический тег
title_meta	VARCHAR(250)	заголовок окна в браузере, тег Title
anons	TEXT	анонс
anons_plus	ENUM('0', '1')	добавлять анонс к описанию: 0 - нет, 1 - да
text	TEXT	описание
sort	INT(11)	порядковый номер для сортировки

timeedit	INT(10)	время последнего изменения в формате UNIXTIME
access	ENUM('0', '1')	доступ ограничен: 0 - нет, 1 - да
admin_id	INT(11)	пользователь из таблицы {users} , добавивший или первый отредктировавший фотографию в административной части
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}

Альбомы фотографий

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
name	TEXT	название
act	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 для sitemap.xml
priority	VARCHAR(3)	Priority для sitemap.xml
noindex	ENUM('0','1')	не индексировать: 0 - нет, 1 - да
parent_id	INT(11)	идентификатор родителя из таблицы {photo_category}
count_children	SMALLINT(5)	количество вложенных альбомов
site_id	INT(11)	идентификатор страницы сайта из таблицы {site}
keywords	VARCHAR(250)	ключевые слова, тег Keywords
descr	TEXT	описание, тэг Description
canonical	VARCHAR(100)	канонический тег
title_meta	VARCHAR(250)	заголовок окна в браузере, тег Title
anons	TEXT	анонс
anons_plus	ENUM('0', '1')	добавлять анонс к описанию: 0 - нет, 1 - да
text	TEXT	описание
sort	INT(11)	порядковый номер для сортировки
timeedit	INT(10)	время последнего изменения в формате UNIXTIME
access	ENUM('0', '1')	доступ ограничен: 0 - нет, 1 - да
admin_id	INT(11)	пользователь из таблицы {users} , добавивший или первый отредктировавший альбом в административной части
theme	VARCHAR(50)	шаблон страницы сайта
view	VARCHAR(50)	шаблон модуля
view_rows	VARCHAR(50)	шаблон модуля для элементов в списке

view_element	VARCHAR(50)
trash	ENUM('0', '1')

категории

шаблон модуля для элементов в категории

запись удалена в корзину: 0 - нет, 1 - да

{photo_category_parents}

Родительские связи альбомов фотографий

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
element_id	INT(11)	идентификатор альбома из таблицы {photo_category}
parent_id	INT(11)	идентификатор альбома-родителя из таблицы {photo_category}
trash	ENUM('0', '1')	запись удалена в корзину: 0 - нет, 1 - да

{photo_category_rel}

Связи фотографий с альбомами

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
element_id	INT(11)	идентификатор фотографии из таблицы {photo}
cat_id	INT(11)	идентификатор альбома из таблицы {photo_category}
trash	ENUM('0', '1')	запись удалена в корзину: 0 - нет, 1 - да

{photo_counter}

Счетчик просмотров фотографий

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
element_id	INT(11)	идентификатор фотографии из таблицы {photo}
count_view	INT(11)	количество просмотров
trash	ENUM('0', '1')	запись удалена в корзину: 0 - нет, 1 - да

{photo_rel}

Связи похожих фотографий

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
element_id	INT(11)	идентификатор фотографии из таблицы {photo}
rel_element_id	INT(11)	идентификатор связанной фотографии из таблицы {photo}
trash	ENUM('0', '1')	запись удалена в корзину: 0 - нет, 1 - да

{postman}

Уведомления

Поле	Тип (размерность)	Описание
master_id	INT(10)	главный идентификатор в формате UNIXTIME
slave_id	INT(11)	вторичный идентификатор в интервале master_id
id	TEXT	объединенный идентификатор
type	ENUM('mail', 'sms')	тип уведомления: mail - электронные письма, sms - короткие сообщения
recipient	TEXT	получатель/получатели
subject	TEXT	тема письма
body	TEXT	содержание письма
from	TEXT	адрес отправителя
auto	ENUM('0', '1')	метод отправки уведомления: 0 - ручной, 1 - автоматический
timeedit	INT(10)	время последнего изменения в формате UNIXTIME
timesent	INT(10)	время отправки уведомления в формате UNIXTIME
status	ENUM('0', '1', '2')	отчет об отправке уведомления: 0 - не отправлено, 1 - отправлено, 2 - ошибка отправления
error	TEXT	отчет об ошибке отправления
trace	TEXT	трассировка отправления

{rating}

Рейтинг элементов модулей

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
element_id	INT(11)	идентификатор элемента модуля
module_name	VARCHAR(50)	название модуля
element_type	ENUM('element', 'cat')	тип элемента модуля
rating	DOUBLE	средняя оценка
count_votes	INT(11)	количество оценок
created	INT(10)	дата последней оценки
trash	ENUM('0', '1')	запись удалена в корзину: 0 - нет, 1 - да

{redirect}

Редиректы

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
redirect	VARCHAR(255)	исходная псевдоссылка, с которой установлен редирект
code	SMALLINT(5)	код редиректа
module_name	VARCHAR(50)	название модуля
element_id	INT(11)	идентификатор элемента модуля

element_type	ENUM('element', 'cat', 'brand', 'param')	тип элемента модуля
trash	ENUM('0', '1')	запись удалена в корзину: 0 - нет, 1 - да

{reviews}

Отзывы

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
user_id	INT(11)	идентификатор пользователя из таблицы {users}
act	ENUM('0', '1')	показывать на сайте: 0 - нет, 1 - да
readed	ENUM('0', '1')	сообщение прочитано: 0 - нет, 1 - да
votes	DOUBLE	средняя оценка
count_votes_plus	INT(11)	количество согласных
count_votes_minus	INT(11)	количество несогласных
created	INT(10)	дата создания
module_name	VARCHAR(50)	название модуля
element_type	ENUM('element', 'cat')	тип элемента модуля
element_id	INT(11)	идентификатор элемента модуля
text	TEXT	ответ
trash	ENUM('0', '1')	запись удалена в корзину: 0 - нет, 1 - да

{reviews_param}

Дополнительные поля отзывов

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
name	VARCHAR(250)	название
type	VARCHAR(20)	тип
info	VARCHAR(20)	смысловая нагрузка
module_name	VARCHAR(50)	название модуля
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
text	TEXT	описание
config	TEXT	сериализованные настройки поля
trash	ENUM('0', '1')	запись удалена в корзину: 0 - нет, 1 - да

{reviews_param_element}

Значения дополнительных полей отзывов

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
value	TEXT	значение
param_id	INT(11)	идентификатор характеристики из таблицы {reviews_param}
element_id	INT(11)	идентификатор отзыва из таблицы {reviews}
trash	ENUM('0', '1')	запись удалена в корзину: 0 - нет, 1 - да

{reviews_param_select}

Варианты значения дополнительных полей отзывов с типом список

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
param_id	INT(11)	идентификатор характеристики из таблицы {reviews_param}
name	VARCHAR(50)	значение
value	VARCHAR(1)	значение для типа характеристики «галочка»: 0 - нет, 1 - да
sort	INT(11)	порядковый номер для сортировки
trash	ENUM('0', '1')	запись удалена в корзину: 0 - нет, 1 - да

{rewrite}

Псевдоссылки

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
rewrite	VARCHAR(250)	псевдоссылка
module_name	VARCHAR(50)	название модуля
element_id	INT(11)	идентификатор элемента модуля
element_type	ENUM('element', 'cat', 'brand', 'param')	тип элемента модуля
trash	ENUM('0', '1')	запись удалена в корзину: 0 - нет, 1 - да

{search_history}

История поисковых запросов

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
created	INT(10)	дата создания
name	TEXT	поисковый запрос

{search_index}

Связи слов и проиндексированных для поиска элементов

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
keyword_id	INT(11)	идентификатор слова из таблицы

result_id	INT(11)
rating	TINYINT(2)

[{search_keywords}](#)

идентификатор проиндексированного элемента из таблицы [{saerch_results}](#)
рейтинг для сортировки результатов

{search_keywords}

Индексированные для поиска слова

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
keyword	VARCHAR(255)	уникальное слово

{search_results}

Индексированные для поиска элементы

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
element_id	INT(11)	идентификатор элемента
table_name	VARCHAR(50)	таблица элемента
lang_id	TINYINT(2)	идентификатор языка сайта из таблицы {languages}
access	ENUM('0', '1')	доступ к элементу ограничен
rating	TINYINT(2)	рейтинг для сортировки результатов
date_start	INT(10)	дата начала показа
date_finish	INT(10)	дата окончания показа

{sessions}

Сессии

Поле	Тип (размерность)	Описание
user_id	INT(11)	идентификатор пользователя из таблицы {users}
session_id	VARCHAR(64)	уникальный 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}

Контрольные хэши авторизованных пользователей

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
user_id	INT(11)	идентификатор пользователя из таблицы {users}
hash	CHAR(32)	хеш
created	INT(10)	дата создания

{shop}

Товары

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
name	TEXT	название
act	ENUM('0', '1')	показывать на сайте: 0 - нет, 1 - да
date_start	INT(10)	дата начала показа
date_finish	INT(10)	дата окончания показа
article	VARCHAR(30)	артикул
measure_unit	VARCHAR(50)	единица измерения
weight	VARCHAR(50)	вес
length	VARCHAR(50)	длина
width	VARCHAR(50)	ширина
height	VARCHAR(50)	высота
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 для sitemap.xml
priority	VARCHAR(3)	Priority для sitemap.xml
noindex	ENUM('0','1')	не индексировать: 0 - нет, 1 - да
cat_id	INT(11)	идентификатор основной категории из таблицы {shop_category}
site_id	INT(11)	идентификатор страницы сайта из таблицы {site}
brand_id	INT(11)	идентификатор производителя из таблицы {shop_brand}
keywords	VARCHAR(250)	ключевые слова, тег Keywords
descr	TEXT	описание, тэг Description
canonical	VARCHAR(100)	канонический тег
title_meta	VARCHAR(250)	заголовок окна в браузере, тег Title
anons	TEXT	анонс
anons_plus	ENUM('0', '1')	добавлять анонс к описанию: 0 - нет, 1 - да
text	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)
google	TEXT	данные для выгрузки в Google Merchant
show_google	ENUM('0', '1')	выгружать в Google Merchant: 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(100)	собственный идентификатор товара при импорте
sort	INT(11)	порядковый номер для сортировки
timeedit	INT(10)	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 из таблицы {users} , добавивший или первый отредктировавший товар в административной части
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}

Сопутствующие услуги

Поле	Тип (размерность)	Описание
id	SMALLINT(5)	идентификатор
name	VARCHAR(50)	название
text	TEXT	описание
act	ENUM('0', '1')	показывать на сайте: 0 - нет, 1 - да
shop_rel	ENUM('0', '1')	связана с товаром: 0 - нет, 1 - да
percent	DOUBLE	процент от стоимости товаров в корзине
price	DOUBLE	цена
amount	DOUBLE	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_additional_cost_category_rel}

Связь сопутствующих услуг и категорий

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
cat_id	INT(11)	идентификатор категории из таблицы {shop_category}
element_id	SMALLINT(5)	идентификатор сопутствующей услуги из таблицы {shop_additional_cost}
trash	ENUM('0', '1')	запись удалена в корзину: 0 - нет, 1 - да

{shop_additional_cost_rel}

Связь сопутствующих услуг и товаров

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
element_id	INT(11)	идентификатор товара из таблицы {shop}
additional_cost_id	SMALLINT(5)	идентификатор сопутствующей услуги из таблицы {shop_additional_cost}
summ	DOUBLE	сумма
trash	ENUM('0', '1')	запись удалена в корзину: 0 - нет, 1 - да

{shop_brand}

Бренды

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
name	TEXT	название
act	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 для sitemap.xml
priority	VARCHAR(3)	Priority для sitemap.xml
noindex	ENUM('0', '1')	не индексировать: 0 - нет, 1 - да
site_id	INT(11)	идентификатор страницы сайта из таблицы {site}
keywords	VARCHAR(250)	ключевые слова, тег Keywords
descr	TEXT	описание, тэг Description
canonical	VARCHAR(100)	канонический тег
title_meta	VARCHAR(250)	заголовок окна в браузере, тег Title
text	TEXT	описание
import	ENUM('0', '1')	производитель только что импортирован: 0 - нет, 1 - да
import_id	VARCHAR(100)	собственный идентификатор производителя при импорте
sort	INT(11)	порядковый номер для сортировки
timeedit	INT(10)	время последнего изменения в формате UNIXTIME
admin_id	INT(11)	пользователь из таблицы {users} , добавивший или первый отредактировавший производителя в административной части
theme	VARCHAR(50)	шаблон страницы сайта
view	VARCHAR(50)	шаблон модуля
trash	ENUM('0', '1')	запись удалена в корзину: 0 - нет, 1 - да

{shop_brand_category_rel}

Связи производителей и категорий

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
element_id	INT(11)	идентификатор производителя из таблицы {shop_brand}
cat_id	INT(11)	идентификатор категории из таблицы {shop_category}
trash	ENUM('0', '1')	запись удалена в корзину: 0 - нет, 1 - да

{shop_cart}

Товары в корзине

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 создания
count	DOUBLE	количество товара
param	TEXT	сериализованные данные о характеристиках товара (доступных к выбору при заказе)
additional_cost	TEXT	идентификаторы сопутствующих услугах, разделенные запятой
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}

Категории товаров

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
name	TEXT	название
act	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 для sitemap.xml
priority	VARCHAR(3)	Priority для sitemap.xml
noindex	ENUM('0', '1')	не индексировать: 0 - нет, 1 - да
parent_id	INT(11)	идентификатор родителя из таблицы {shop_category}
count_children	SMALLINT(5)	количество вложенных категорий
site_id	INT(11)	идентификатор страницы сайта из таблицы {site}
keywords	VARCHAR(250)	ключевые слова, тег Keywords

descr	TEXT	описание, тэг Description
canonical	VARCHAR(100)	канонический тег
title_meta	VARCHAR(250)	заголовок окна в браузере, тег Title
anons	TEXT	анонс
anons_plus	ENUM('0', '1')	добавлять анонс к описанию: 0 - нет, 1 - да
text	TEXT	описание
show_yandex	ENUM('0', '1')	выгружать в Яндекс.Маркет: 0 - нет, 1 - да (если в настройках выбрана выгрузка только выбранных категорий)
show_google	ENUM('0', '1')	выгружать в Google Merchant: 0 - нет, 1 - да (если в настройках выбрана выгрузка только выбранных товаров)
import	ENUM('0', '1')	категория только что импортирован: 0 - нет, 1 - да
import_id	VARCHAR(100)	собственный идентификатор категории при импорте
sort	INT(11)	порядковый номер для сортировки
timeedit	INT(10)	время последнего изменения в формате UNIXTIME
access	ENUM('0', '1')	доступ ограничен: 0 - нет, 1 - да
admin_id	INT(11)	пользователь из таблицы {users} , добавивший или первый отредактировавший категорию в административной части
theme	VARCHAR(50)	шаблон страницы сайта
view	VARCHAR(50)	шаблон модуля
view_rows	VARCHAR(50)	шаблон модуля для элементов в списке категории
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}

Родительские связи категорий товаров

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}

Связи товаров и категорий

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
element_id	INT(11)	идентификатор товара из таблицы {shop}

cat_id	INT(11)
trash	ENUM('0', '1')

идентификатор категории из таблицы [{shop_category}](#)

запись удалена в корзину: 0 - нет, 1 - да

{shop_counter}

Счетчик просмотров товаров

Поле	Тип (размерность)
id	INT(11)
element_id	INT(11)
count_view	INT(11)
trash	ENUM('0', '1')

Описание

идентификатор

идентификатор товара из таблицы [{shop}](#)

количество просмотров

запись удалена в корзину: 0 - нет, 1 - да

{shop_currency}

Дополнительные валюты магазина

Поле	Тип (размерность)
id	SMALLINT(5)
name	VARCHAR(20)
exchange_rate	DOUBLE
trash	ENUM('0', '1')

Описание

идентификатор

название

курс к основной валюте

запись удалена в корзину: 0 - нет, 1 - да

{shop_delivery}

Способы доставки

Поле	Тип (размерность)
id	SMALLINT(5)
name	VARCHAR(50)
text	TEXT
service	VARCHAR(50)
params	TEXT
act	ENUM('0', '1')
sort	INT(11)
trash	ENUM('0', '1')

Описание

идентификатор

название

описание

служба доставки

сериализованные настройки службы доставки

показывать на сайте: 0 - нет, 1 - да

порядковый номер для сортировки

запись удалена в корзину: 0 - нет, 1 - да

{shop_delivery_thresholds}

Стоимость способов доставки

Поле	Тип (размерность)
id	SMALLINT(5)
delivery_id	SMALLINT(5)
price	DOUBLE
amount	DOUBLE
trash	ENUM('0', '1')

Описание

идентификатор

идентификатор способа доставки из таблицы [{shop_delivery}](#)

стоимость

сумма, от которой действует стоимость

запись удалена в корзину: 0 - нет, 1 - да

{shop_discount}

Скидки

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
date_start	INT(10)	дата начала действия
date_finish	INT(10)	дата окончания действия
discount	DOUBLE	процент скидки
amount	DOUBLE	действует от цены товара
deduction	DOUBLE	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 из таблицы {users_role} , для которого установлена скидка
act	ENUM('0', '1')	показывать на сайте: 0 - нет, 1 - да
person	ENUM('0', '1')	скидка действует только для определенных пользователей: 0 - нет, 1 - да
text	TEXT	описание скидки для администратора
trash	ENUM('0', '1')	запись удалена в корзину: 0 - нет, 1 - да

{shop_discount_coupon}

Купоны на скидку

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
discount_id	INT(11)	идентификатор скидки из таблицы {shop_discount}
coupon	VARCHAR(10)	код купона
count_use	TINYINT(3)	сколько раз можно использовать купон
used	TINYINT(3)	сколько раз купон использован при оформлении заказа
trash	ENUM('0', '1')	запись удалена в корзину: 0 - нет, 1 - да

{shop_discount_object}

Товары и категории, на которые действуют скидки

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_discount_person}

Пользователи, для которых действуют скидки

Поле	Тип (размерность)	Описание
------	-------------------	----------

id	INT(11)	идентификатор
discount_id	INT(11)	идентификатор скидки из таблицы {shop_discount}
user_id	INT(11)	идентификатор пользователя из таблицы {users}
session_id	VARCHAR(64)	номер сессии пользователя из таблицы {sessions}
coupon_id	INT(11)	идентификатор купона из таблицы {shop_discount_coupon}
used	ENUM('0', '1')	скидка уже использована: 0 - нет, 1 - да
trash	ENUM('0', '1')	запись удалена в корзину: 0 - нет, 1 - да

{shop_files_codes}

Коды для скачивания товаров-нематериальных активов

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}

Описание полей файлов импорта

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
name	VARCHAR(250)	название
cat_id	INT(11)	идентификатор описания файла из таблицы {shop_import}
type	VARCHAR(20)	тип
params	TEXT	сериализованные данные о поле
required	ENUM('0', '1')	сериализованные данные о поле
sort	INT(11)	выдавать ошибку, если поле не заполнено: 0 - нет, 1 - да
trash	ENUM('0', '1')	запись удалена в корзину: 0 - нет, 1 - да

{shop_import_category}

Описание файлов импорта

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
name	VARCHAR(250)	название
format	ENUM('csv', 'xls')	формат файла
type	ENUM('good', 'category', 'brand')	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 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
header	ENUM('0', '1')	первая строка - названия столбцов: 0 - нет, 1 - да
sort	INT(11)	порядковый номер для сортировки
trash	ENUM('0', '1')	запись удалена в корзину: 0 - нет, 1 - да

{shop_order}

Заказы

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	DOUBLE	общая сумма заказа
delivery_id	VARCHAR(10)	способ доставки из таблицы {shop_delivery}
delivery_summ	DOUBLE	стоимость доставки
delivery_info	TEXT	данные службы доставки
discount_id	SMALLINT(5)	идентификатор скидки из таблицы {shop_discount}
discount_summ	DOUBLE	сумма скидки
count_minus	ENUM('0', '1')	товары списаны: 0 - нет, 1 - да
trash	ENUM('0', '1')	запись удалена в корзину: 0 - нет, 1 - да

{shop_order_additional_cost}

Сопутствующие услуги, включенные в заказ

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
order_id	INT(11)	идентификатор заказа из таблицы {shop_order}

order_goods_id	INT(11)
additional_cost_id	SMALLINT(5)
summ	DOUBLE

идентификатор записи о купленном товаре из таблицы [{shop_order_goods}](#)
идентификатор сопутствующей услуги из таблицы [{shop_additional_cost}](#)
сумма

{shop_order_goods}

Товары в заказе

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	DOUBLE	количество товаров
price	DOUBLE	цена
trash	ENUM('0', '1')	запись удалена в корзину: 0 - нет, 1 - да

{shop_order_goods_param}

Дополнительных характеристики товаров в заказе

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
value	INT(11)	значение характеристики
param_id	INT(11)	идентификатор характеристики из таблицы {shop_param}
order_goods_id	INT(11)	идентификатор записи о купленном товаре из таблицы {shop_order_goods}
trash	ENUM('0', '1')	запись удалена в корзину: 0 - нет, 1 - да

{shop_order_param}

Поля конструктора формы оформления заказа

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
name	VARCHAR(250)	название
text	TEXT	описание
type	VARCHAR(20)	тип
info	VARCHAR(30)	смысловая нагрузка
sort	INT(11)	порядковый номер для сортировки
required	ENUM('0', '1')	обязательно для заполнения: 0 - нет, 1 - да
show_in_form	ENUM('0', '1')	использовать в стандартной форме оформления заказа: 0 - нет, 1 - да
show_in_form_register	ENUM('0', '1')	позволять редактировать из личного кабинета: 0 - нет, 1 - да

show_in_form_one_click	ENUM('0', '1')	использовать в форме быстрого заказа: 0 - нет, 1 - да
config	TEXT	дополнительные настройки поля
trash	ENUM('0', '1')	запись удалена в корзину: 0 - нет, 1 - да

{shop_order_param_element}

Значения полей конструктора оформления заказа

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
value	TEXT	значение
param_id	INT(11)	идентификатор 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}

Варианты значений полей конструктора оформления заказа типа список

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
param_id	INT(11)	идентификатор поля из таблицы {shop_order_param}
value	TINYINT(2)	значение для типа характеристики «галочка»: 0 - нет, 1 - да
name	VARCHAR(50)	значение
sort	INT(11)	порядковый номер для сортировки
trash	ENUM('0', '1')	запись удалена в корзину: 0 - нет, 1 - да

{shop_order_param_user}

Значения полей конструктора оформления заказа, предзаполненные пользователями

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
value	TEXT	значение
param_id	INT(11)	идентификатор 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}

Статусы заказов

Поле	Тип (размерность)	Описание
id	TINYINT(3)	идентификатор
name	VARCHAR(50)	название

color	VARCHAR(20)	цвет
status	ENUM('0', '1', '2', '3', '4')	действие статуса заказа
sort	INT(11)	порядковый номер для сортировки
count_minus	ENUM('0', '1')	списывать товары: 0 - нет, 1 - да
send_mail	ENUM('0', '1')	отправлять уведомление пользователю о смене статуса: 0 - нет, 1 - да
trash	ENUM('0', '1')	запись удалена в корзину: 0 - нет, 1 - да

{shop_param}

Дополнительные характеристики товаров

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
name	VARCHAR(250)	название
type	VARCHAR(20)	тип
sort	INT(11)	порядковый номер для сортировки
search	ENUM('0', '1')	характеристика выводится в форме поиска: 0 - нет, 1 - да
list	ENUM('0', '1')	характеристика выводится в списке товаров: 0 - нет, 1 - да
block	ENUM('0', '1')	характеристика выводится в блоке товаров: 0 - нет, 1 - да
id_page	ENUM('0', '1')	характеристика выводится на странице товара: 0 - нет, 1 - да
required	ENUM('0', '1')	доступен к выбору при заказе: 0 - нет, 1 - да
page	ENUM('0', '1')	отдельная страница для значений: 0 - нет, 1 - да
text	TEXT	описание
config	TEXT	дополнительные настройки поля
display_in_sort	ENUM('0', '1')	выводится в блоке для сортировки: 0 - нет, 1 - да
yandex_use	ENUM('0', '1')	выгружается в файл YML: 0 - нет, 1 - да
yandex_name	VARCHAR(100)	название в файле YML
yandex_unit	VARCHAR(50)	единица измерения в файле YML
measure_unit	VARCHAR(50)	единица измерения
site_id	INT(11)	идентификатор страницы сайта из таблицы {site}
trash	ENUM('0', '1')	запись удалена в корзину: 0 - нет, 1 - да

{shop_param_category_rel}

Связи дополнительных характеристик товаров и категорий

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
element_id	INT(11)	идентификатор характеристики из таблицы {shop_param}

cat_id	INT(11)	идентификатор категории из таблицы {shop_category}
trash	ENUM('0', '1')	запись удалена в корзину: 0 - нет, 1 - да

{shop_param_element}

Значения дополнительных характеристик товаров

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
value	TEXT	значение
param_id	INT(11)	идентификатор характеристики 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}

Варианты значений дополнительных характеристик товаров типа список

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
param_id	INT(11)	идентификатор характеристики из таблицы {shop_param}
value	TINYINT(2)	значение для типа характеристики «галочка»: 0 - нет, 1 - да
name	VARCHAR(50)	значение
sort	INT(11)	порядковый номер для сортировки
trash	ENUM('0', '1')	запись удалена в корзину: 0 - нет, 1 - да

{shop_price}

Цены товаров

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
good_id	INT(11)	идентификатор товара из таблицы {shop}
price	DOUBLE	цена
old_price	DOUBLE	исходная цена
count_goods	DOUBLE	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	DOUBLE	скидка
discount_id	INT(11)	идентификатор скидки из таблицы {shop_discount}
person	ENUM('0', '1')	цена действует только для определенных пользователей: 0 - нет, 1 - да
role_id	INT(11)	идентификатор типа пользователя из

таблицы [{users_role}](#), для которого действует скидка

идентификатор валюты из таблицы [{shop_currency}](#)

собственный идентификатор при импорте
запись удалена в корзину: 0 - нет, 1 - да

currency_id	SMALLINT(5)
import_id	VARCHAR(100)
trash	ENUM('0', '1')

{shop_price_image_rel}

Изображения товаров, прикрепленные к цене

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
price_id	INT(11)	идентификатор исходной цены 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}

Дополнительные характеристики, учитываемые в цене

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
price_id	INT(11)	идентификатор исходной цены из таблицы {shop_price}
param_id	INT(11)	идентификатор характеристики из таблицы {shop_param}
param_value	INT(11)	значение характеристики - идентификатор из таблицы {shop_param_select}
trash	ENUM('0', '1')	запись удалена в корзину: 0 - нет, 1 - да

{shop_rel}

Связи похожих товаров

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}

Товары в списке ожидания

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)	е-mail пользователя
created	INT(10)	дата создания
param	TEXT	сериализованные данные о характеристиках товара (доступных к выбору при заказе)
trash	ENUM('0', '1')	запись удалена в корзину: 0 - нет, 1 - да

{shop_wishlist}

Товары в списке 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	VARCHAR(64)	номер сессии пользователя из таблицы {sessions}
created	INT(10)	дата создания
count	DOUBLE	количество товара
param	TEXT	сериализованные данные о характеристиках товара (доступных к выбору при заказе)
additional_cost	TEXT	идентификаторы сопутствующих услуг, разделенные запятой
is_file	ENUM('0', '1')	товар-файл: 0 - нет, 1 - да
trash	ENUM('0', '1')	запись удалена в корзину: 0 - нет, 1 - да

{site}

Страницы сайта

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
parent_id	INT(11)	идентификатор родителя из таблицы {site}
count_children	SMALLINT(5)	количество вложенных страниц
name	VARCHAR(100)	название
title_meta	VARCHAR(250)	заголовок окна в браузере, тег Title
keywords	VARCHAR(250)	ключевые слова, тег Keywords
descr	TEXT	описание, тэг Description
canonical	VARCHAR(100)	канонический тег
text	LONGTEXT	контент
act	ENUM('0', '1')	показывать на сайте: 0 - нет, 1 - да
access	ENUM('0', '1')	доступ ограничен: 0 - нет, 1 - да
date_start	INT(10)	дата начала показа
date_finish	INT(10)	дата окончания показа
admin_id	INT(11)	пользователь из таблицы {users} , добавивший или первый отредактировавший страницу в

title_no_show	ENUM('0', '1')	административной части
map_no_show	ENUM('0', '1')	не копировать заголовок в H1: 0 - нет, 1 - да
changefreq	ENUM('always', 'hourly', 'daily', 'weekly', 'monthly', 'yearly', 'never')	Changefreq для sitemap.xml
priority	VARCHAR(3)	Priority для sitemap.xml
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
sort	INT(11)	порядковый номер для сортировки
timeedit	INT(10)	время последнего изменения в формате UNIXTIME
theme	VARCHAR(50)	шаблон страницы сайта
module_name	VARCHAR(50)	прикрепленный модуль
js	TEXT	JS-код
trash	ENUM('0', '1')	запись удалена в корзину: 0 - нет, 1 - да

{site_blocks}

Блоки на сайте

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
name	VARCHAR(100)	название
text	TEXT	описание
act	ENUM('0', '1')	показывать на сайте: 0 - нет, 1 - да
access	ENUM('0', '1')	доступ ограничен: 0 - нет, 1 - да
title_no_show	ENUM('0', '1')	не показывать заголовок: 0 - нет, 1 - да
date_start	INT(10)	дата начала показа
date_finish	INT(10)	дата окончания показа
admin_id	INT(11)	пользователь из таблицы {users} , добавивший или первый отредактировавший блок в административной части
sort	INT(11)	порядковый номер для сортировки
timeedit	INT(10)	время последнего изменения в формате UNIXTIME
trash	ENUM('0', '1')	запись удалена в корзину: 0 - нет, 1 - да

{site_blocks_site_rel}

Данные о том, на каких страницах выводятся блоки на сайте

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
element_id	INT(11)	идентификатор блока из таблицы {site_blocks}
site_id	INT(11)	идентификатор страницы сайта из таблицы {site}

trash	ENUM('0', '1')
-------	----------------

запись удалена в корзину: 0 - нет, 1 - да

{site_dynamic}

Динамические блоки

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
name	VARCHAR(100)	название
text	TEXT	описание
act	ENUM('0', '1')	показывать на сайте: 0 - нет, 1 - да
type	VARCHAR(20)	тип
access	ENUM('0', '1')	доступ ограничен: 0 - нет, 1 - да
title_no_show	ENUM('0', '1')	не показывать заголовок: 0 - нет, 1 - да
date_start	INT(10)	дата начала показа
date_finish	INT(10)	дата окончания показа
admin_id	INT(11)	пользователь из таблицы {users} , добавивший или первый отредактировавший блок в административной части
sort	INT(11)	порядковый номер для сортировки
timeedit	INT(10)	время последнего изменения в формате UNIXTIME
trash	ENUM('0', '1')	запись удалена в корзину: 0 - нет, 1 - да

{site_dynamic_element}

Контент динамических блоков, заполненный в элементах модулей

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
dynamic_id	INT(11)	идентификатор динамического блока из таблицы {site_dynamic}
module_name	VARCHAR(50)	название модуля
element_id	INT(11)	идентификатор элемента модуля
element_type	VARCHAR(20)	тип элемента
value	TEXT	значение
parent	ENUM('0', '1')	применить к элементам категории: 0 - нет, 1 - да
category	ENUM('0', '1')	применить к вложенным элементам: 0 - нет, 1 - да
trash	ENUM('0', '1')	запись удалена в корзину: 0 - нет, 1 - да

{site_dynamic_module}

Данные о том, в каких модулях заполняются динамические блоки

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
dynamic_id	INT(11)	идентификатор динамического блока из таблицы {site_dynamic}

module_name	VARCHAR(50)	название модуля
element_type	VARCHAR(20)	тип элемента
trash	ENUM('0', '1')	запись удалена в корзину: 0 - нет, 1 - да

{site_parents}

Родительские связи страниц сайта

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}

Рассылки

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
created	INT(10)	дата создания
send	ENUM('0', '1')	рассылка отправлена: 0 - нет, 1 - да
cat_id	INT(11)	идентификатор категории из таблицы {subscription_category}
name	VARCHAR(250)	название
text	TEXT	текст рассылки
trash	ENUM('0', '1')	запись удалена в корзину: 0 - нет, 1 - да

{subscription_category}

Категории рассылок

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
name	VARCHAR(250)	название
act	ENUM('0', '1')	показывать на сайте: 0 - нет, 1 - да
sort	INT(11)	порядковый номер для сортировки
text	TEXT	описание
parent_id	INT(11)	идентификатор родителя из таблицы {subscription_category}
count_children	SMALLINT(5)	количество вложенных категорий
trash	ENUM('0', '1')	запись удалена в корзину: 0 - нет, 1 - да

{subscription_category_parents}

Родительские связи категорий рассылок

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
element_id	INT(11)	идентификатор категории из таблицы {subscription_category}

parent_id	INT(11)
trash	ENUM('0', '1')

идентификатор родителя категории из таблицы [{subscription_category}](#)
запись удалена в корзину: 0 - нет, 1 - да

{subscription_category_rel}

Связи рассылок и категорий

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}

Подписчики на рассылку

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
created	INT(10)	дата добавления
name	VARCHAR(250)	имя
mail	VARCHAR(64)	e-mail
code	VARCHAR(32)	код доступа к управлению подпиской
act	ENUM('0', '1')	получает рассылку: 0 - нет, 1 - да
trash	ENUM('0', '1')	запись удалена в корзину: 0 - нет, 1 - да

{subscription_emails_cat_unrel}

Отключенные категории рассылок у подписчиков

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
element_id	INT(11)	идентификатор подписчика из таблицы {subscription_emails}
cat_id	INT(11)	идентификатор категории из таблицы {subscription_category}
trash	ENUM('0', '1')	запись удалена в корзину: 0 - нет, 1 - да

{subscription_phones}

Телефоны для рассылок

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
created	INT(10)	дата добавления
name	VARCHAR(250)	имя
phone	VARCHAR(64)	телефон
act	ENUM('0', '1')	получает рассылку: 0 - нет, 1 - да
trash	ENUM('0', '1')	запись удалена в корзину: 0 - нет, 1 - да

{subscription_sms}

SMS-рассылки

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
created	INT(10)	дата создания
send	ENUM('0', '1')	рассылка отправлена: 0 - нет, 1 - да
name	VARCHAR(250)	название
text	TEXT	текст
trash	ENUM('0', '1')	запись удалена в корзину: 0 - нет, 1 - да

{tags}

Связи тегов и элементов модулей

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
module_name	VARCHAR(50)	название модуля
element_id	INT(11)	идентификатор элемента модуля
element_type	ENUM('element', 'cat')	тип элемента модуля
tags_name_id	INT(11)	идентификатор тега из таблицы {tags_name}
act	ENUM('0', '1')	показывать на сайте: 0 - нет, 1 - да
access	ENUM('0', '1')	доступ к элементу ограничен
date_start	INT(10)	дата начала показа
date_finish	INT(10)	дата окончания показа
trash	ENUM('0', '1')	запись удалена в корзину: 0 - нет, 1 - да

{tags_name}

Теги

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
name	VARCHAR(70)	тег
map_no_show	ENUM('0', '1')	не показывать на карте сайта: 0 - нет, 1 - да
sort	INT(11)	порядковый номер для сортировки
changefreq	ENUM('always', 'hourly', 'daily', 'weekly', 'monthly', 'yearly', 'never')	Changefreq для sitemap.xml
priority	VARCHAR(3)	Priority для sitemap.xml
noindex	ENUM('0', '1')	не индексировать: 0 - нет, 1 - да
keywords	VARCHAR(250)	ключевые слова, тег Keywords
descr	TEXT	описание, тэг Description
title_meta	VARCHAR(250)	заголовок окна в браузере, тег Title
text	TEXT	описание
timeedit	INT(10)	время последнего изменения в формате UNIXTIME
trash	ENUM('0', '1')	запись удалена в корзину: 0 - нет, 1 - да

{trash}

Корзина с удаленными элементами

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
table_name	VARCHAR(50)	название таблицы удаленного элемента
module_name	VARCHAR(50)	название модуля удаленного элемента
element_id	INT(11)	идентификатор удаленного элемента
created	INT(10)	дата удаления
parent_id	INT(11)	идентификатор родителя из таблицы {trash}
count_children	SMALLINT(5)	количество удаленных вместе с текущей записей
user_id	INT(11)	идентификатор пользователя из таблицы {users}

{trash_parents}

Родительские связи в корзине (какое удаление влечет за собой другое)

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
element_id	INT(11)	идентификатор записи из таблицы {trash} об удалении элемента
parent_id	INT(11)	идентификатор записи-родителя из таблицы {trash} об удалении элемента

{update_return}

Точки возврата

Поле	Тип (размерность)	Описание
id	SMALLINT(5)	идентификатор
name	VARCHAR(100)	название
created	INT(10)	дата создания
current	ENUM('0', '1')	текущая точка: 0 - нет, 1 - да
hash	VARCHAR(100)	уникальный хэш
text	TEXT	описание
version	VARCHAR(32)	версия сборки

{users}

Пользователи

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
name	VARCHAR(60)	логин
password	VARCHAR(32)	пароль в зашифрованном виде
mail	VARCHAR(64)	e-mail
phone	VARCHAR(64)	телефон
created	INT(10)	дата регистрации

fio	VARCHAR(255)	ФИО
role_id	TINYINT(3)	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
copy_files	ENUM('0', '1')	сохранять картинки с внешних сайтов, при вставке контента в визуальный редактор: 0 - нет, 1 - да
useradmin	ENUM('0', '1', '2')	панель быстрого редактирования: 0 - отключена, 1 - включена, 2 - только панель без режима редактирования
start_admin	VARCHAR(30)	стартовая страница административной части
lang_id	TINYINT(2)	идентификатор языка сайта из таблицы {languages}
admin_nastr	SMALLINT(5)	количество элементов на странице в административной части
identity	VARCHAR(255)	URL на страницу в соц. сети
config	TEXT	Настройки пользователя
trash	ENUM('0', '1')	запись удалена в корзину: 0 - нет, 1 - да

{users_actlink}

Код активации аккаунта

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
link	VARCHAR(255)	код активации
user_id	INT(11)	идентификатор пользователя из таблицы {users}
created	INT(10)	дата создания
count	INT(11)	количество неудачных попыток

{users_param}

Дополнительные поля с данными о пользователях

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
name	VARCHAR(250)	название
text	TEXT	описание
type	VARCHAR(20)	тип
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}

Значения дополнительных полей с данными о пользователях

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
value	TEXT	значение
param_id	INT(11)	идентификатор 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}

Связи дополнительных полей с данными о пользователях и типов пользователей

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
element_id	INT(11)	идентификатор 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}

Варианты значений дополнительных полей с данными о пользователях для типа список

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
param_id	INT(11)	идентификатор поля из таблицы {users_param}
value	TINYINT(2)	значение для типа характеристики «галочка»: 0 - нет, 1 - да
name	VARCHAR(50)	название
sort	INT(11)	порядковый номер для сортировки
trash	ENUM('0', '1')	запись удалена в корзину: 0 - нет, 1 - да

{users_role}

Типы пользователей

Поле	Тип (размерность)	Описание
id	TINYINT(3)	идентификатор
name	VARCHAR(64)	название
registration	ENUM('0', '1')	задавать при регистрации: 0 - нет, 1 - да

only_self	ENUM('0', '1')
sort	INT(11)
trash	ENUM('0', '1')

видеть только свои материалы: 0 - нет, 1 - да
 порядковый номер для сортировки
 запись удалена в корзину: 0 - нет, 1 - да

{users_role_perm}

Права типов пользователей

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
role_id	SMALLINT(5)	идентификатор типа пользователя из таблицы {users_role}
perm	TEXT	права на действия: all, init, edit, del
rewrite	TEXT	тег доступа (например, название модуля)
type	ENUM('site', 'admin')	часть сайта: site - пользовательская, admin - административная

{visitors_session}

Статистика посетителей

Поле	Тип (размерность)	Описание
master_id	INT(10)	главный идентификатор в формате UNIXTIME
slave_id	INT(11)	вторичный идентификатор в интервале master_id
id	TEXT	объединенный идентификатор
session_id	VARCHAR(64)	идентификатор сессии из таблицы {sessions}
user_id	INT(11)	идентификатор пользователя из таблицы {users}
role_id	INT(11)	тип пользователя из таблицы {users_role} , для которого установлена скидка
hostname	VARCHAR(128)	хост
user_agent	TEXT	браузер пользователя
create	VARCHAR(10)	время создания в формате UNIXTIME
timestamp	VARCHAR(20)	время, до которого сессия действует
timeedit	VARCHAR(10)	время последнего изменения в формате UNIXTIME
status	ENUM('0', '1')	статус валидации пользовательского агента: 0 - бот, 1 - посетитель
search_bot	VARCHAR(255)	имя поискового бота

{visitors_stat_traffic}

Статистика посещений

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
date	INT(10)	время в формате UNIXTIME
visits_count	INT(11)	количество визитов
visits_search_bot	INT(11)	количество визитов от поисковых ботов

_count		
visits_bot_count	INT(11)	количество визитов от неопределенных ботов
pageviews_count	INT(11)	количество просмотров
pageviews_search_bot_count	INT(11)	количество просмотров от поисковых ботов
pageviews_bot_count	INT(11)	количество просмотров от неопределенных ботов
users_count	INT(11)	количество уникальных посетители
users_search_bot_count	INT(11)	количество уникальнх поисковых ботов
users_bot_count	INT(11)	количество уникальных неопределенных ботов

{visitors_stat_traffic_names_search_bot}

Статистика по поисковым ботам

Поле	Тип (размерность)	Описание
master_id	INT(10)	главный идентификатор в формате UNIXTIME
slave_id	INT(11)	вторичный идентификатор в интервале master_id
id	TEXT	объединенный идентификатор
search_bot	VARCHAR(255)	имя поискового бота
visits_search_bot_count	INT(11)	количество визитов от поисковых ботов
timeedit	VARCHAR(10)	время последнего изменения в формате UNIXTIME

{visitors_stat_traffic_pages}

Статистика по посещаемым страницам сайта

Поле	Тип (размерность)	Описание
master_id	INT(10)	главный идентификатор в формате UNIXTIME
slave_id	INT(11)	вторичный идентификатор в интервале master_id
id	TEXT	объединенный идентификатор
site_id	INT(11)	идентификатор страницы сайта из таблицы {site}
visits_count	INT(11)	количество визитов
visits_search_bot_count	INT(11)	количество визитов от поисковых ботов
visits_bot_count	INT(11)	количество визитов от неопределенных ботов
timeedit	VARCHAR(10)	время последнего изменения в формате UNIXTIME

{visitors_stat_traffic_source}

Статистика по источникам трафика

Поле	Тип (размерность)	Описание
------	-------------------	----------

master_id	INT(10)	главный идентификатор в формате UNIXTIME
slave_id	INT(11)	вторичный идентификатор в интервале master_id
id	TEXT	объединенный идентификатор
referer_domain	TEXT	домен-рефферер
visits_count	INT(11)	количество визитов
visits_search_bot_count	INT(11)	количество визитов от поисковых ботов
visits_bot_count	INT(11)	количество визитов от неопределенных ботов
timeedit	VARCHAR(10)	время последнего изменения в формате UNIXTIME

{visitors_url}

Лог посещения пользователей

Поле	Тип (размерность)	Описание
master_id	INT(10)	главный идентификатор в формате UNIXTIME
slave_id	INT(11)	вторичный идентификатор в интервале master_id
id	TEXT	объединенный идентификатор
visitors_session_id	TEXT	идентификатор посетителей из таблицы {visitors_session}
hostname	VARCHAR(128)	хост
referer_scheme	ENUM('0', '1')	протокол рефферер: 0 - http, 1 - https
referer_domain	TEXT	домен-рефферер
referer_rewrite	TEXT	адрес-рефферер
referer_query	TEXT	параметры-рефферер
scheme	ENUM('0', '1')	протокол: 0 - http, 1 - https
rewrite	TEXT	псевдоссылка страницы
query	TEXT	параметры адреса
is_admin	ENUM('0', '1')	страница относится к административной части сайта: 0 - нет, 1 - да
site_id	INT(11)	идентификатор страницы сайта из таблицы {site}
module_name	VARCHAR(50)	название модуля
element_id	INT(11)	идентификатор элемента модуля
element_type	ENUM('element', 'cat', 'brand', 'param')	тип элемента модуля
is_mobile	ENUM('0', '1')	использование мобильного устройства: 0 - нет, 1 - да
is_mobile_url	ENUM('0', '1')	мобильная версия страницы: 0 - нет, 1 - да
user_id	INT(11)	идентификатор пользователя из таблицы {users}
role_id	INT(11)	тип пользователя из таблицы {users_role} , для которого установлена скидка

visits	ENUM('0', '1' , '2')	посещение: 0 - нет, 1 - новое, 2 - повторное
status	ENUM('0', '1')	статус валидации пользовательского агента: 0 - бот, 1 - посетитель
search_bot	VARCHAR(255)	имя поискового бота
timeedit	VARCHAR(10)	время последнего изменения в формате UNIXTIME

{votes}

Опросы

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
name	TEXT	название
act	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_answers}

Варианты ответов на опросы

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
text	TEXT	описание
votes_id	INT(11)	идентификатор вопроса из таблицы {votes}
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_site_rel}

Данные о том, на каких страницах сайта выводятся опросы

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
element_id	INT(11)	идентификатор вопроса из таблицы {votes}
site_id	INT(11)	идентификатор страницы сайта из таблицы {site}
trash	ENUM('0', '1')	запись удалена в корзину: 0 - нет, 1 - да

{votes_userversion}

Варианты ответов пользователей

Поле	Тип (размерность)	Описание
id	INT(11)	идентификатор
votes_id	INT(11)	идентификатор вопроса из таблицы {votes}
text	VARCHAR(250)	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. В нем определяется какие функции в модели подключить, какие шаблонные теги используются в модуле.
- *modules/news/news.model.php* – модель модуля, в которой описаны функции модуля. В модели запрашиваются данные из БД, идет их обработка и готовится переменная-массив *\$result* для передачи информации в шаблон модуля.
- Файлы в папке *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, то добавляется файл:

- [обработка](#) – файл *modules/имя_модуля/имя_модуля.action.php* класс *Имя_модуля_action*.

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.

Отдельно следует отметить [JS-скрипты модуля](#). С ними тоже связаны некоторые особенности.

Контроллер

- [Класс Controller](#)

Контроллер модуля – класс *Модуль* в файле *modules/модуль/модуль.php*.

В классе описаны:

- массив `$rewrite_variable_names` – переменные, передаваемые в URL;
- функция `init()` – инициализация модуля;
- функция `action()` – обработка данных в модуле;
- шаблонные функции.

Переменные, передаваемых в URL определяются для каждого модуля ([список доступных переменных](#)).

Пример:

```
public $rewrite_variable_names = array('page', 'show', 'year', 'month', 'day', 'param', 'edit', 'sort');
```

Функция инициализации модуля вызывается, если модуль подключен к текущей странице. В ней происходит выбор функции модели модуля в зависимости от переданных параметров. Модель модуля подключается автоматически при первом вызове ее из контроллера через переменную `$this->module`.

Пример:

```
// если в URL передана переменная show, вызываем функцию id в модели – показываем
// страницу товара
if($this->diafan->_route->show)
{
    $this->model->id();
}
// иначе вызываем функцию list_ в модели – выводим список товаров
else
{
    $this->model->list_();
}
```

Функция обработки данных вызывается, если передана переменная `$_POST["module"] = 'название_модуля'`. Обработка действий модуля подключается автоматически при первом вызове ее из контроллера через переменную `$this->action`.

В этой функции можно

- проверить авторизован ли пользователь:

Пример:

```
$this->action->check_user();  
  
if ($this->action->result())  
    return;
```

- проверить идентификационный хэш авторизованного пользователя:

Пример:

```
$this->action->check_user_hash();  
  
if ($this->action->result())  
    return;
```

- убедиться, что запрос был сделан со страницы модуля

Пример:

```
if($this->diafan->_site->module != 'forum')  
    return;
```

- выбрать функцию из файла *modules/модуль/модуль.action.php*, которая будет обрабатывать запрос:

Пример:

```
if(! empty($_POST["action"]))  
{  
    switch($_POST["action"])  
    {  
        case 'add':  
            return $this->action->add();  
  
        case 'edit':  
            return $this->action->edit();  
    }  
}
```

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

var `rewrite_variable_names` = array() – переменные, передаваемые в URL страницы.

Пример:

```
// в модуле будет использована пагинация, деление на категории и показана страница
элемента
public $rewrite_variable_names = array('page', 'cat', 'show');
```

var **result** – сгенерированные в модели данные, передаваемые в шаблон.

Методы

void **init ()** – Инициализация модуля.

Функция определяется в контроллере модуля.

Пример:

```
// в модуле "Файловый архив" modules/files/files.php
public function init()
{
    if($this->diafan->configmodules("cat"))
    {
        $this->rewrite_variable_names[] = 'cat';
    }

    if ($this->diafan->_route->show)
    {
        $this->model->id();
    }
    elseif (! $this->diafan->configmodules("cat"))
    {
        $this->model->list_();
    }
    elseif (! $this->diafan->_route->cat)
    {
        $this->model->first_page();
    }
    else
    {
        $this->model->list_category();
    }
}
```

void **action ()** – Обрабатывает полученные данные из формы.

Функция определяется в контроллере модуля.

Пример:

```
// в модуле "Корзина товаров" modules/cart/cart.php
public function action()
{
```

```

if($this->diafan->configmodules('security_user', 'shop'))
{
    $this->action->check_user();

    if ($this->action->result())
        return;
}
if(! empty($_POST["action"]))
{
    switch($_POST["action"])
    {
        case 'recalc':
            return $this->action->recalc();

        case 'order':
            return $this->action->order();

        case 'one_click':
            return $this->action->one_click();

        case 'upload_image':
            return $this->action->upload_image();

        case 'delete_image':
            return $this->action->delete_image();
    }
}
}

```

void **show_module ()** – Выводит шаблон модуля.

Шаблон модуля подключается автоматически. Названием шаблона задается в модели переменной `$this->result["view"]`.

void **get_global_variables ()** – Определяет свойства страницы, если они заданы в модуле.

Функция вызывается автоматически.

array **get_attributes (array \$attributes)** – Задаёт неопределённым атрибутам шаблонного тега значение по умолчанию.

- array \$attributes: массив определённых атрибутов

Пример:

```

// шаблонный тег show_block module="clauses" предусматривает несколько атрибутов
// все они перед использованием должны быть переданы как аргументы
// для функции get_attributes в начале функции show_block()
$attributes = $this->get_attributes($attributes, 'count', 'site_id', 'cat_id', 'sort', 'images',
'images_variation', 'only_module', 'template');

```

boolean **method_exists (string \$module, string \$name, [string \$method_name = false])** –

Проверяет, существует ли класс или метод в классе.

- `string $module`: имя модуля
- `string $name`: тип класса
- `string $method_name`: имя метода

Модель

- [Класс Model](#)

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*.

Свойства

var **result** – сгенерированные в модели данные, передаваемые в шаблон.

Методы

boolean **access** (integer \$element_id, [string \$module_name = ''], [string \$element_type = 'element']) – Проверяет есть ли доступ к элементу/категории модуля.

- integer \$element_id: номер элемента
- string \$module_name: модуль
- string \$element_type: тип данных (*element* – элемент (по умолчанию), *cat* – категория)

Если при вызове опустить аргумент *\$module_name*, 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:

```
// если доступ к статье ограничен, то выдаем ошибку 403 Доступ запрещен.
if (! empty($row['access']) && ! $this->access($row['id'], 'clauses'))
{
    Custom::inc('includes/403.php');
}
```

string **format_date** (integer \$date, [string \$module_name = ''], [integer \$site_id = 0]) – Форматирует дату в соответствии с конфигурацией модуля.

- 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/courses/courses.model.php
// конвертация из UNIX-формата даты создания статьи в модуле «Статьи»
$row['date'] = $this->format_date($row['created']);
```

array **get_author** (integer \$user_id) – Получает имя, никнейм и аватар пользователя сайта.

- integer \$user_id: 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:

```
$user = $this->get_author(2);
print_r($user);
/* выведет:
Array
(
    [id] => 2
    [fio] => Михаил Волков
    [name] => michael
    [identity] =>
    [avatar] => http://site.ru/userfiles/avatar/michael.png
    [avatar_width] => 50
    [avatar_height] => 50
    [user_page] => http://site.ru/user/?name=michael
)

или

Гость
*/
```

void **prepare_author** (integer \$author) – Запоминает идентификаторы пользователей сайта, информация о которых понадобится.

- integer \$author: идентификатор автора

Пример:

```
// в этом примере будет выполнено три SQL-запроса к базе данных
// для получения информации о трех пользователях
$ids = array(3, 5, 7);
foreach($ids as $id)
{
    $users[$id] = $this->get_author($id);
}
```

Пример:

```
// в этом примере будет выполнен один SQL-запрос к базе данных
// для получения информации о трех пользователях
$ids = array(3, 5, 7);
foreach($ids as $id)
{
    $this->prepare_author($id);
}
foreach($ids as $id)
{
    $users[$id] = $this->get_author($id);
}
```

array **get_params (array \$config)** – Получает массив полей формы.

- 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:

```
// получение дополнительных полей при обработке формы модуля «Обратная связь»
$rows = $this->get_params(array("module" => "feedback", "where" => "site_id=".$site_id));
```

array **get_breadcrumb ()** – Генерирует данные для навигации "Хлебные крошки".

Пример:

```
// modules/ab/ab.model.php
// использования метода в модуле «Объявления»
$this->result["path"] = $this->get_breadcrumb();
```

boolean **validate_attribute_site_cat (string \$module_name, array &\$site_ids, array &\$cat_ids, array &\$minus)** – Валидация атрибутов cat_id и site_id для шаблонных тегов.

- string \$module_name: название модуля
- array \$site_ids: страница сайта
- array \$cat_ids: категория
- array \$minus: страницы сайта и категории, которые вычитаются

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 **error_insert_tag (string \$error, string \$module_name)** – Выводит ошибку на сайте.

- string \$error: описание ошибки
- string \$module_name: название модуля

void **theme_view ()** – Определяет шаблоны страницы и модуля для элемента.

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.

Пример:

```
/* modules/clauses/clauses.model.php  
Используется при выводе первой страницы (first page) */  
$this->theme_view_first_page();
```

void **theme_view_cat (array \$row)** – Определяет шаблоны страницы и модуля для категории.

- 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.

- 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 МЕТА-тегов элемента.

- 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.

- 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.

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, `{photo_counter}` – для элементов фотогалереи.

void **is_admin ()** – Проверяет является ли текущий пользователь администратором.

Пример:

```
if($this->is_admin())  
{  
    echo 'Здравствуйте, администратор!';  
}
```

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:

```
echo $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:

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

Пример:

```
/* вывод шаблона кнопки «Купить» shop.view.buy_form  
из шаблона страницы товара shop.view.id */  
echo $this->get('buy_form', 'shop', array("row" => $result, "result" => $result));
```

В шаблоне можно выводить шаблонный тег с помощью [функции htmleditor](#).

Пример:

```
// выводим шаблонным тегом динамический блок  
// в шаблоне modules/news/views/news.view.id.php  
echo $this->htmleditor('<insert name="show_dynamic" module="site" id="1">');
```

Обработка POST-запросов

- [Класс Action](#)
- [Методы](#)

Все данные из формы, а также данные, отправленные Ajax-запросом, обрабатываются классом *Модуль_action*, описанным в файле *modules/модуль/модуль.action.php*.

Обработка подключается всегда, когда передана переменная `$_POST["module"] = 'модуль';`

Доступ к этому файлу идет через контроллер *modules/модуль/модуль.php*. Если в контроллере описана функция `action()`, то она определяет какую функцию в файле *modules/модуль/модуль.action.php* в зависимости от того, что передано в переменной `$_POST["action"]`. Если функция `action()` не описана в контроллере, то вызывается функция `init()` в файле *modules/модуль/модуль.action.php*.

Данные модулю могут быть переданы даже если он не подключен ни к одной странице сайта. Проверка на соответствие прикрепленного к странице модуля и вызванного модуля проводится при обработке запроса, если это необходимо.

Структура класса может быть любой для того чтобы реализовывать предусмотренный функционал.

Класс Action

Класс *Модуль_action* наследует класс *Action*, описанный в файле *includes/action.php*.

Методы

Свойства

var **result** – полученный после обработки данных результат.

Методы

void **end ()** – Отправляет ответ.

Функция вызывается из *includes/init.php* после любого вызова обработки действия в модуле.

Пример:

```
$module = new Feedback_action($this->diafan);  
$module->action();  
$module->action->end();
```

boolean **result ()** – Проверяет сформирован ли ответ.

Пример:

```
if(empty($_POST["mail"]))
```

```

{
    $this->result["errors"]["mail"] = "Введите e-mail.";
}

// если ответ сформирован (например, найдена ошибка в передаваемых данных),
// останавливаем выполнение функции обработки
if ($this->result())
    return;

```

void **check_user ()** – Проверяет авторизован ли пользователь на сайте.

Пример:

```

// проверяем авторизован ли пользователь
$this->check_user();

// если найдена ошибка, останавливаем выполнение функции обработки
if ($this->result())
    return;

```

void **check_user_hash ()** – Проверяет хэш пользователя.

Пример:

```

// проверяем верен ли хэш пользователя
$this->check_user_hash();

// если найдена ошибка, останавливаем выполнение функции обработки
if ($this->result())
    return;

```

void **empty_required_field (array \$config)** – Проверяет на заполнение обязательных полей.

- 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:

```

// получаем поля формы обратной связи
$params = $this->model->get_params(array("module" => "feedback", "where" => "site_id=".
    $site_id));

// проверяем заполнены ли обязательные поля
$this->empty_required_field(array("params" => $params));

// если найдена ошибка, останавливаем выполнение функции обработки
if ($this->result())

```

```
return;
```

void **insert_values (array \$config)** – Добавляет значение полей формы в базу данных.

- array \$config: настройки функции: id номер элемента, table таблица, params поля формы, multilang значения переводятся

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.action.php
// запись данных при оформлении заказа в модуле «Корзина»
$this->insert_values(array("id" => $order_id, "table" => "shop_order", "params" => $params));
```

boolean **check_site_id ()** – Проверяет корректность номера страницы сайта..

Пример:

```
// проверяем правильно ли передан номер страницы сайта,
// к которой подключен модуль
$this->check_site_id();

// если найдена ошибка, останавливаем выполнение функции обработки
if ($this->result())
    return;
```

void **check_captcha ()** – Проверяет правильность капчи.

При вызове функции она обращается к [методам класса «Captcha»](#) для проверки правильности ввода. Если капча введена неверно, будет возвращено сообщение с ошибкой.

Пример:

```
// проверяем правильно ли введен защитный код
$this->check_site_id();

// если найдена ошибка, останавливаем выполнение функции обработки
if ($this->result())
    return;
```

void **update_values (array \$config)** – Обновляет значение полей формы в базу данных.

- array \$config: настройки функции: id номер элемента, table таблица, params поля формы, prefix префикс, multilang значения переводятся, no_empty_param_ids массив не требующих заполнения параметров

Пример:

```
// modules/usersettings/usersettings.action.php
// изменение данных текущего пользования при редактировании профиля через
личный кабинет
$this->update_values(array("id" => $this->diafan->_users->id, "table" => "users", "params" =>
$params));
```

boolean **valid_email (string \$email, string \$field)** – Проверка e-mail на валидность.

- 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.action.php
// проверка правильности формата введенной электронной почты при отправке
вопроса; модуль «Вопрос-ответ»
$this->valid_email($_POST['email'], "email");

// если найдена ошибка, останавливаем выполнение функции обработки
if ($this->result())
    return;
```

boolean **valid_phone (string \$phone, string \$field)** – Проверка телефона на валидность.

- string \$phone: телефон
- string \$field: название поля в массиве \$_POST

Пример:

```
// modules/registration/registration.action.php
// проверка правильности формата введенного телефона при регистрации
пользователя
$this->valid_phone($_POST['phone'], "phone");

// если найдена ошибка, останавливаем выполнение функции обработки
if ($this->result())
    return;
```

boolean **valid_url (string \$url, string \$field)** – Проверка ссылки на валидность.

- string \$url: ссылка
- string \$field: название поля в массиве \$_POST

array **get_attributes (array \$attributes)** – Задает неопределенным атрибутам шаблонного тега значение по умолчанию.

- array \$attributes: массив определенных атрибутов

void **action ()** – Обрабатывает полученные данные из формы.

void **action ()** – Обрабатывает полученные данные из формы.

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 любого места в DIAFAN.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)
    {
        $row = DB::query_fetch_array("SELECT [name], [anons] FROM {news} WHERE id=%d AND
[act]='1' AND trash='0'", $id);

        return $row;
    }
}
```

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 нем должен быть определен класс *Модуль_install*, который наследует класс *Install*.

Файл инсталляции используется при установке DIAFAN.CMS, при установке / удалении модуля из административной части, при восстановлении базы данных из административной части.

Свойства

var **is_core** = false – модуль является частью ядра.

Пример:

```
// modules/site/site.install.php
public $is_core = true;
```

var **module** – название текущего модуля.

Свойство заполняется автоматически при инициации установки модуля и содержит название модуля.

Пример:

При удалении модуля удаляется папка загружаемых файлов модуля. Название модуля получаем из свойства *module*, так как функционал в *includes/install.php* общий для всех модулей.

```
// includes/install.php функция uninstall()
if(is_dir(ABSOLUTE_PATH.USERFILES.'/'.$this->module))
{
    File::delete_dir(USERFILES.'/'.$this->module);
}
```

var **title** – название.

Пример:

```
// modules/users/users.install.php
public $title = "Пользователи";
```

var **install_modules** – установленные/устанавливаемые модули.

Содержит массив со всеми установленными или устанавливаемыми в текущий момент модулями. Используется, например, для заполнения примеров в модулях, которые связаны с другими модулями.

Пример:

При создании модуля создаем папку shop в папке модуля, если модуль «Магазин» тоже устанавливается.

```
// modules/my/my.install.php функция action()
if(in_array('shop', $this->install_modules))
{
    File::create_dir(USERFILES.'/my/shop');
}
```

var **langs** – идентификаторы языков сайта.

Содержит массив идентификаторов языков сайта на момент установки модуля.

Пример:

При создании модуля добавляем изображение печати из демо-данных для каждого языка сайта.

```
// modules/my/my.install.php функция action()
foreach($this->langs as $lang_id)
{
    File::copy_file(DEMO_PATH.'my/stamp'.$lang_id.'.png', USERFILES.'/my/stamp'.
$lang_id.'.png');
}
```

var **tables = array()** – таблицы в базе данных.

Формат:

```
public $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,
                // комментарий виден из админера базы данных
                "comment" => "комментарий к полю",
            ),
            ...
        ),
        // ключи таблицы
        "keys" => array(
            // пример PRIMARY KEY (id) или KEY site_id (`site_id`)
```

```

        "описание_ключа",
        ...
    ),
    // комментарий виден из админера базы данных
    "comment" => "комментарий к таблице",
),
...
);

```

Пример:

Создаем таблицу категорий при установке модуля «Новости».

```

public $tables = array(
    array(
        "name" => "news_category",
        "comment" => "Категории новостей",
        "fields" => array(
            array(
                "name" => "id",
                "type" => "INT(11) UNSIGNED NOT NULL AUTO_INCREMENT",
                "comment" => "идентификатор",
            ),
            array(
                "name" => "name",
                "type" => "TEXT",
                "comment" => "название",
                "multilang" => true,
            ),
            array(
                "name" => "act",
                "type" => "ENUM('0', '1') NOT NULL DEFAULT '0'",
                "comment" => "показывать на сайте: 0 - нет, 1 - да",
                "multilang" => true,
            ),
            array(
                "name" => "site_id",
                "type" => "INT(11) UNSIGNED NOT NULL DEFAULT '0'",
                "comment" => "идентификатор страницы сайта из таблицы {site}",
            ),
            array(
                "name" => "text",
                "type" => "TEXT",
                "comment" => "описание",
                "multilang" => true,
            ),
            array(
                "name" => "trash",
                "type" => "ENUM('0', '1') NOT NULL DEFAULT '0'",
                "comment" => "запись удалена в корзину: 0 - нет, 1 - да",
            ),
        ),
    ),
);

```

```

    ),
),
"keys" => array(
    "PRIMARY KEY (id)",
    "KEY site_id (site_id)",
),
)
);

```

Все таблицы модуля должны начинаться с названия модуля.

Пример:

Таблицы модуля «Новости» – {*news*}, {*news_category*} и т. д.

var **modules = array()** – записи в таблице {modules}.

При установке модуля обязательно нужно сделать запись в таблице [{modules}](#), чтобы система определила модуль как установленный.

Формат:

```

public $modules = array(
    array(
        // пример: cart
        "name" => "название_подмодуля",
        // 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: Корзина товаров
        // можно опустить, если запись name равна названию основного модуля
        "title" => "название_модуля_на_русском",
    ),
);

```

Пример:

Запись в таблице {modules} для модуля «Файловый архив».

```

// modules/files/files.install.php
public $modules = array(
    array(
        "name" => "files",
        "admin" => true,
        "site" => true,
        "site_page" => true,
    ),
);

```

```
),  
);
```

var **admin** = **array()** – меню административной части.

Формат:

```
public $admin = 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,  
            ),  
        ),  
    ),  
    ...  
);
```

Пример:

Создаем страницы управления модулем «Обратная связь».

```
// modules/feedback/feedback.install.php  
public $admin = array(  
    array(  
        "name" => "Обратная связь",  
        "rewrite" => "feedback",  
        "group_id" => 2,  
        "sort" => 19,  
        "act" => true,  
        "docs" => "http://www.diafan.ru/moduli/obratnaya_svyaz/",  
        "children" => array(  
            array(  
                "name" => "Сообщения",  
                "rewrite" => "feedback",  
                "act" => true,  
            ),  
        ),  
    ),  
);
```

```

    ),
    array(
        "name" => "Конструктор формы",
        "rewrite" => "feedback/param",
        "act" => true,
    ),
    array(
        "name" => "Настройки",
        "rewrite" => "feedback/config",
    ),
)
);

```

var **site = array()** – страницы сайта.

Формат:

```

public $site = array(
    array(
        // пример: Каталог товаров, Shop
        "name" => array("название_на_русском", "название_на_английском"),
        // пример: 15 – необязательный параметр
        "sort" => "порядковый_номер_для_сортировки",
        // пример: shop – необязательный параметр
        "module_name" => "прикрепленный_модуль",
        // пример: catalog-tovarov
        "rewrite" => "ЧПУ",
        // пример: sitemap.php – необязательный 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 – необязательный параметр
        "menu" => "номер_категории_меню",
    ),
    ...
);

```

Пример:

Создаем страницу сайта «Обратная связь» при установке модуля.

```

// modules/feedback/feedback.install.php
public $site = array(

```

```

    array(
        "parent_id" => 4,
        "name" => array('Обратная связь', 'Feedback'),
        "act" => true,
        "module_name" => "feedback",
        "rewrite" => "feedback",
        "menu" => 1,
    ),
);

```

var **config = array()** – настройки.

Формат:

```

public $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",
    ),
);

```

Пример:

В магазине отмечаем настройку «Использовать категории» при установке.

```

// modules/shop/shop.install.php
public $config = array(
    array(
        "name" => "cat",
        "value" => "1",
    )
);

```

var **sql = array()** – предустановленные данные.

Заполняется, если при установке модуля нужно вставить начальные или обязательные данные в таблицы базы данных модуля.

Формат:

```

public $sql = array(
    "название_таблицы" => array(
        array(
            "поле" => "значение",
            "поле2" => "значение2",
        ),
        array(

```



```

        "поле" => "значение",
        "поле2" => "значение2",
    ),
),
"название_таблицы2" => array(
    ...
),
...
);

```

Пример:

В модуле «Меню» создаем предустановленные категории.

```

// modules/menu/menu.install.php
public $sql = array(
    "menu_category" => array(
        array(
            'id' => 1,
            'name' => array('Меню верхнее', 'Top menu'),
            'current_link' => '1',
            'show_all_level' => '1',
        ),
        array(
            'id' => 2,
            'name' => array('Меню интернет-магазин', 'Shop menu'),
        ),
    ),
);

```

var **demo = array()** – демо-данные.

Формат тот же, что у предыдущего свойства. Но данные заполняются только в том случае, если отмечена опция «Заполнить сайт примерами из демо-версии».

Пример:

В модуле «Новости» добавляем категорию для демонстрации возможностей модуля.

```

// modules/news/news.install.php
public $demo = array(
    'news_category' => array(
        array(
            'id' => 1,
            'name' => array('Новости компании', 'Company news'),
            'rewrite' => 'news/novosti-kompanii',
        )
    )
);

```

Методы

void **start (boolean \$demo)** – Устанавливаем модуль.

- boolean \$demo: установить демо-данные

Пример:

```
// устанавливаем модуль «Новости»
include_once(ABSOLUTE_PATH.'modules/news/news.install.php');
$module = News_install($this->diafan);
$module->start(true);
```

void **action ()** – Выполняет действия при установке модуля.

Функция может быть определена, если перед установкой нужно совершить действия, невыполняемые в стандартной логике установки.

Пример:

В модуле «Языки сайта» дополнительный язык добавляется, если отмечена опция «Создать две языковые версии сайта».

```
// modules/languages/languages.install.php
protected function action()
{
    if (count($this->langs) > 1)
    {
        $this->sql["languages"][] =
            array(
                "id" => 2,
                "name" => 'eng',
                "shortname" => 'eng',
            );
    }
}
```

void **action_post ()** – Выполняет действия при установке модуля после основной установки.

Функция может быть определена, если нужно выполнить набор действий после установки всех модулей.

Пример:

В модуле «Поиск по сайту» нужно проиндексировать контент всех модулей.

```
// modules/search/search.install.php
public function action_post()
{
    $this->diafan->_search->index_all();
}
```

void **tables** ([array \$array = array()]) – Добавляет таблицы.

- array \$array: таблицы

void **modules** ([array \$array = array()]) – Добавляет запись в таблицу {modules}.

- array \$array: массив с данными

void **admin** ([array \$array = array()], [integer \$parent_id = 0], [array \$parent = array()]) – Добавляет записи о модуле в таблицу {admin} - страницы админки.

- array \$array: массив с данными
- integer \$parent_id: номер страницы админки - родителя
- array \$parent: значения полей для родителя

void **site** ([array \$rows = array()]) – Добавляет страницы сайта.

- array \$rows: массив с данными

void **config** ([array \$array = array()]) – Добавляет запись в таблицу {config}.

- array \$array: массив с данными

void **sql** ([array \$array = array()]) – Выполняет SQL-запросы.

- array \$array: массив с данными

void **demo** () – Установка демо-данных.

void **uninstall** () – Удаляет модуль.

void **uninstall_action** () – Выполняет действия при удалении модуля.

Функция может быть определена, если нужно выполнить набор действий после удаления модуля.

Пример:

В модуле «Интернет магазин» нужно удалить создаваемую страницу «Заказ оформлен», к которой модуль не подключен.

```
// modules/shop/shop.install.php
protected function uninstall_action()
{
    DB::query("DELETE FROM {rewrite} WHERE rewrite='shop/cart/done'");
    DB::query("DELETE FROM {site} WHERE [name]='Заказ оформлен'");
}
```

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/payment/payment.get.php* – *http://site.ru/payment/get/* – 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.

В стандартной поставке DIAFAN.CMS есть три модуля, которые нуждаются в файлах-настройках в модулях, которых их используют:

- [Карта сайта](#) требует файл *modules/модуль/модуль.sitemap.php*;
- [Поиск](#) требует файл *modules/модуль/модуль.search.php*;
- [Меню на сайте](#) требует файл *modules/модуль/admin/модуль.admin.menu.php*.

Формат файла заложен в подключаемом модуле.

JS-скрипты

- [Автоматическое подключение](#)
- [Ручное подключение.](#)
- [Стандартная обработка Ajax-запросов](#)

Для работы модуля часто нужен отдельный JavaScript-сценарий. Есть два пути их подключения:

Автоматическое подключение

Файл *modules/модуль/js/модуль.js* подключается автоматически к странице с прикрепленным модулем. Файл *modules/модуль/js/модуль.шаблон.js* подключается автоматически к шаблону модуля.

Пример:

К шаблону *modules/shop/views/shop.view.buy_form.php* автоматически подгрузиться файл *modules/shop/js/shop.buy_form.js*.

Файл *modules/shop/js/shop.js* подгрузиться на всех страницах модуля (если модуль подключен к странице сайта), но не подгрузиться к шаблонным тегам модуля.

Файлы не дублируются. Например, файл *modules/shop/js/shop.buy_form.js* подключится один раз, даже если кнопка «Купить» будет выведена несколько раз.

Ручное подключение.

Можно указать непосредственно скрипт, который нужно подключить. Сделать это можно в любом месте: в шаблоне или в модели модуля.

Пример:

```
// внутренняя ссылка
$this->diafan->_site->js_view[] = 'modules/search/js/search.show_search.js';
// внешняя ссылка
$this->diafan->_site->js_view[] =
'http://www.google.com/recaptcha/api/js/recaptcha_ajax.js';
```

Все JS-файлы модулей собираются пока генерируются страница и выводятся шаблонным тегом `show_js`. Все локальные файлы сжимаются, некоторые файлы подгружаются асинхронно. Важно, чтобы шаблонный тег `<insert name="show_js">`, подключающий JS-файлы, был в конце HTML-документа. Перед закрывающим тегом `</body>`.

Стандартная обработка Ajax-запросов

В DIAFAN.CMS предусмотрена отправка данных формы Ajax-запросом, если у формы есть атрибут `class="ajax"`. Ответ принимаются данные в формате JSON, которые всегда одинаково обрабатываются. Стандартно обработаются следующие данные (задается в

modules/модуль/модуль.action.php).

Пример:

```
// произойдет редирект на главную
$this->result["redirect"] = '/';

// сообщения подставляются в специальные поля
// общее сообщение добавиться в контейнер <div class="error"></div> обычно под
кнопкой <input type="submit">
$this->result["errors"][0] = 'Общее сообщение для всей формы';
// сообщение для конкретного поля добавиться в контейнер <div
class="error_name"></div>
$this->result["errors"]["name"] = 'Введите имя';

$this->result["data"] = array(
    // "данные" подставятся в блок <div class="target"></div>
    ".target" => "данные",
    // "data" подставятся в блок <div id="target2"></div>
    "#target2" => "data",
    // "текст вместо формы" заменит форму
    "form" => "текст вместо формы",
    // блок <div class="target3"></div> будет скрыт
    ".target3" => false,
);

// обновиться капча
$this->result["captcha"] = 'код капчи';

// очистит форму
$this->result["result"] = "success";

// добавит данные в конец блока `<div class="значение_атрибута_id_формы"></div>`
$this->result["add"] = "данные";

// загрузить загруженные файлы
$this->result["attachments"] = "блок загруженных файлов";

// загрузить загруженные изображения
$this->result["images"] = "блок загруженных изображений";

// обновит идентификационный хэш пользователя
$this->result["hash"] = "новый хэш";
```

В JS-файле модуля можно дописать свою обработку для действия перед отправкой формы и при получении результатов. Эти действия дополняют стандартную обработку, если функция вернула **true**, или заменят её, если функция вернула **false**. Формат:

```
diafan_ajax.before['метка_формы'] = function(form){}
```

```
diafan_ajax.success['метка_формы'] = function(form, response){}
```

Метка формы состоит из названия модуля и действия формы. То есть из содержимого полей формы:

```
<input name="module">
```

```
<input name="action">
```

Пример:

```
// перед отправкой формы поиска объявления функция смотрит есть ли на странице
контейнер <div class="ab_list"></div>
// если есть, то данные отправляются Ajax-запросом
diafan_ajax.before['ab_search'] = function(form){
    if(! $(".ab_list").length)
    {
        $(form).removeClass('ajax').submit();
        return false;
    }
    $(form).attr('method', 'POST');
}

// когда ответ пришел, то данные помещаются в контейнер <div class="ab_list"></div>
// и отменяется стандартная обработка
diafan_ajax.success['ab_search'] = function(form, response){
    var k = 0;
    $(".ab_list").text("");
    $(".ab_list").first().html(response.data).focus();
    return false;
}
```


CSS-файлы

- [Автоматическое подключение](#)
- [Ручное подключение.](#)

Для модуля можно подключить отдельный CSS-файл (начиная с версии 6.0.6.2). Подключаются файлы также как модульные JS-скрипты.

Автоматическое подключение

Файл *modules/модуль/css/модуль.css* подключается автоматически к странице с прикрепленным модулем. Файл *modules/модуль/css/модуль.шаблон.css* подключается автоматически к шаблону модуля.

Пример:

К шаблону *modules/shop/views/shop.view.buy_form.php* автоматически подгрузиться файл стилей *modules/shop/css/shop.buy_form.css*.

Файл *modules/shop/css/shop.css* подгрузиться на всех страницах модуля (если модуль подключен к странице сайта), но не подгрузиться к шаблонным тегам модуля.

Ручное подключение.

Можно указать непосредственно скрипт, который нужно подключить. Сделать это можно в любом месте: в шаблоне или в модели модуля.

Пример:

```
$this->diafan->_site->css_view[] = 'modules/search/css/search.show_search.css';
```

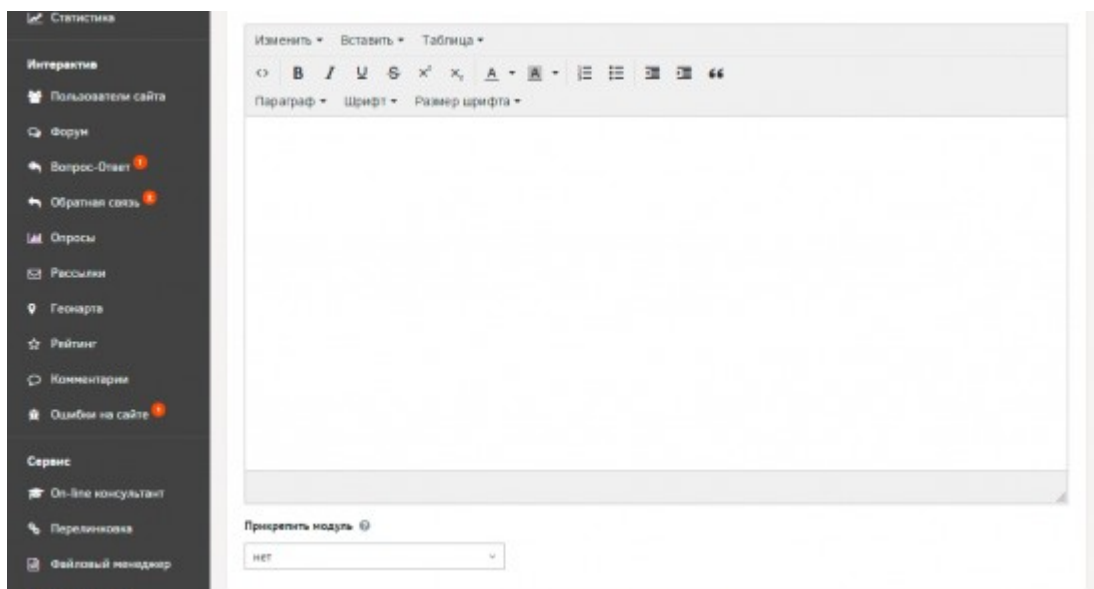
CSS-файлы выводятся в начале страницы с помощью шаблонного тега [show_css](#). Если тег не используется или если файлы подключен уже после вывода тега, то остальные CSS-файлы будут выведены тегом [show_js](#) внизу страницы.

Все локальные файлы сжимаются, если не включен режим разработки.

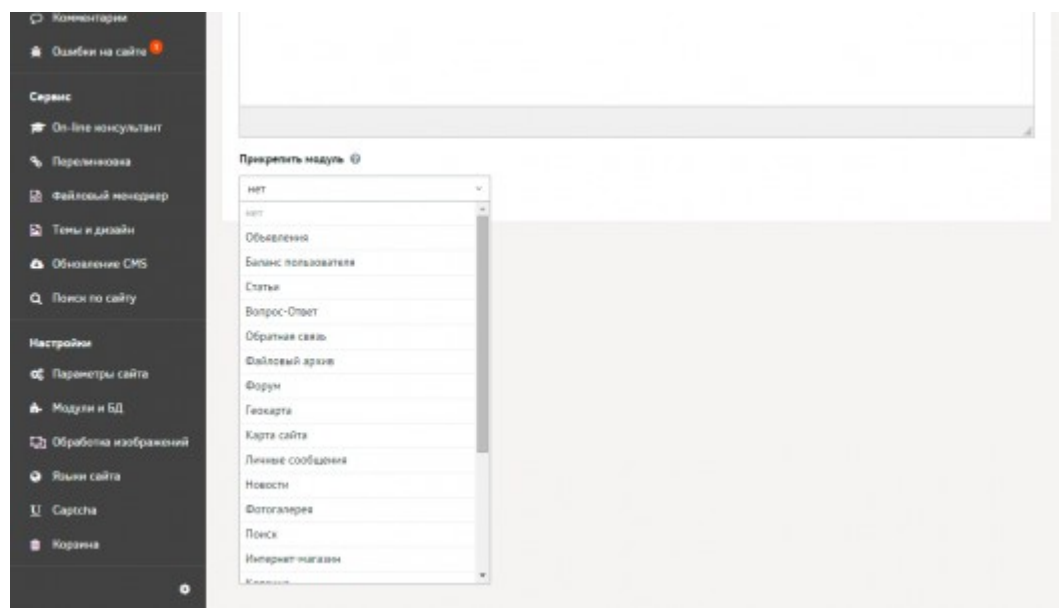
Подключение модуля

Подключаемый модуль выбирается из списка модулей при редактировании страницы сайта.

Шаг 1. Открываем страницу на редактирование



Шаг 2. Выбираем модуль из выпадающего списка



Модуль будет доступен в списке, если в базе данных в таблице `{modules}` есть запись `name=Название_модуля, module_name=Название_основного_модуля, site='1', site_page='1', title=Название_модуля_по_русски`. Эта запись в таблице делается при установке модуля.

Большинство модулей можно прикрепить к нескольким страницам, при этом контент для различных страниц будет различным. Также для разных страниц можно задать различную конфигурацию модуля.

Административная часть сайта

- [Общие данные](#)
- [Общие функции](#)
- [Объекты классов-подключений](#)

Инициализация административной части похожа на инициализацию пользовательской части сайта. Но имеет свои особенности из-за различия структуры модуля и доступа к данным.

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 пользователя и проверка доступа к административной части;
3. определение языковой версии данных для редактирования;
4. определение 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. общие данные;
2. общие функции;
3. объекты классов-подключений.

Теперь подробнее об этих данных:

Общие данные

- `$this->diafan->installed_modules` – массив данных с установленными модулями;

Пример:

```
if(in_array('search', $this->diafan->installed_modules))
{
    // модуль «Поиск на сайте» установлен и можно к нему обращаться
}
```

- `$this->diafan->rewrite_variable_names` – список всех переменных, доступных для использования в URL;

- `$this->diafan->current_module` – модуль текущего исполняемого файла.

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:

```
// обрезаем текст до 50 символов, сохраняя полными слова на конце
$short_text = $this->diafan->short_text($text, 50);
```

Объекты классов-подключений

В основном объекте иницируются классы:

- кеширование (класс `Cache`, файл `includes/cache.php`, вызывается `$this->diafan->_cache`) – вся работа с кэшем;
- маршрутизатор (класс `Route`, файл `includes/route.php`, вызывается `$this->diafan->_route`) – доступ к переменным, передаваемым в URL, а также генерирование, сохранение, получение ЧПУ;
- шаблоны модулей (класс `Template`, файл `includes/template.php`, вызывается `$this->diafan->_tpl`) – подключение шаблонов модулей, обработка текста из визуального редактора;
- парсер шаблонных тегов (класс `Parser_theme`, файл `includes/parser_theme.php`, вызывается `$this->diafan->_parser_theme`) – подключает шаблонные теги, прописанные в виде тегов в шаблонах сайта и непосредственно в тексте визуального редактора.

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`, создаётся экземпляр класса.

Выделим отдельно подключения модулей, относящихся к ядру системы:

- страницы админки – используется для получения данных о текущей странице в административной части.

Пример:

- `$this->diafan->_admin->rewrite` – ЧПУ текущей страницы;
 - `$this->diafan->_admin->module` – модуль, прикрепленный к текущей странице.
- пользователи сайта – используется для получения сведений о текущем пользователе.

Пример:

- `$this->diafan->_users->id` – id-номер текущего пользователя;
 - `$this->diafan->_users->name` – логин текущего пользователя;
 - `$this->diafan->_users->fio` – ФИО текущего пользователя.
- языки сайта – используется для получения сведений о языках сайта, главной языке административной и пользовательской частей сайта и получение перевода интерфейса.

Пример:

- `$this->diafan->_languages->all` – массив данных обо всех языках, использующихся на сайте;
- `$this->diafan->_languages->site` – номер языка, использующегося как основной на сайте;
- `$this->diafan->_languages->admin` – номер языка, использующегося как основной в административной части.

Основные действия

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 POST запросов](#) – файл *adm/includes/action.php*.

Этих действий обычно достаточно, чтобы управлять большинством наборов данных.

Пример:

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.

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 `list_row()`, формирующую список элементов таблицы модуля.

Пример:

```
public function show()
{
    $this->diafan->list_row();
}
```

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 [\\$variables list](#).

Ссылку «Добавить новый элемент» выводит функция `show_add()`. Обычно эта функция выглядит так:

Пример:

```
public function show_add()
{
    $this->diafan->addnew_init('Добавить новость');
}
```

По каким полям данные будут фильтроваться определено в массиве [\\$variables filter](#).

Редактирование, добавление

Если в URL страницы передана переменная *edit* (<https://site.ru/admin/news/edit5/>) или *addnew* (<https://site.ru/admin/shop/addnew1/>), то подключается функция `edit()` из файла *adm/includes/edit.php*. Функция генерирует форму редактирования на основе массива [\\$variables](#), определенного в классе модуля. Остальные функции в файле *adm/includes/edit.php* детализируют отрисовку формы редактирования.

Другой случай подключения файла *adm/includes/edit.php* – если в файле указано, что это интерфейс редактирования настроек модуля:

```
public $config = array (  
    'config', // файл настроек модуля  
);
```

Любую публичную функцию из файла *adm/includes/edit.php* можно переопределить в файле модуля.

Пример:

Нам понадобилось выводить в подсказке к полю данные из настроек модуля: типы файлов, которые можно загрузить в модуль «Файловый архив». Мы нашли в файле *adm/includes/edit.php* функцию `help()`, которая выводит подсказку к полю.

Мы можем просто ее скопировать в файл *modules/files/admin/files.admin.php* и внести изменения. Можем, чтобы не дублировать код, заменить повтор кода на обращение к родительской функции (из файла *adm/includes/edit.php*).

```
// переопределяем функцию help() в файле modules/files/admin/files.admin.php  
public function help($key = "")  
{  
    // чтобы не дублировать код,  
    // обращаемся к этой же функции из файла adm/includes/edit.php  
    $text = parent::__call('help', array());  
    // добавляем к подсказке типы данных из настроек  
    $text = str_replace('%attachment_extensions', $this->diafan-  
>configmodules('attachment_extensions'), $text);  
    return $text;  
}
```

Также можно дополнить редактирование полей своим представлением вывода. Для этого в классе модуля определяется пользовательская функция `edit_variable_переменная()`. Для файла настроек модуля эта функция будет называться `edit_config_variable_переменная()`.

Пример:

Для заказов в интернет-магазине нам нужно выводить номер заказа без возможности редактирования. А если создается новый заказ, то мы должны показать номер заказа, который будет присвоен.

Так как логика не соответствует ни одному из типов данных (*text*, *numtext*, *editor* и т. д.), определим пользовательскую функцию для вывода номера.

Сначала зададим новое поле в массиве `$variables`.

```
public $variables = array(
    'main' => array(
        'number' => array(
            'type' => 'function',
            'name' => 'Заказ №',
            'help' => 'Номер заказа.',
            'no_save' => true,
        ),
        ...
    ),
    ...
);
```

Тип *function* здесь номинальный. Если пользовательская функция определена, то она подключится к любому типу кроме *module*.

no_save мы указали для того, чтобы поле не сохранялось. Мы используем его только для того, чтобы вывести информацию в форме редактирования.

Теперь определим саму пользовательскую функцию.

```
// modules/shop/admin/shop.admin.order.php
public function edit_variable_number()
{
    // стандартный блок вывода поля в форме
    echo '<div class="unit" id="number">';
    // название поля number, которое мы задаем в массиве $variables
    echo '<b>'. $this->diafan->variable_name(). '</b> ';

    // если запись не новая, просто выводим номер
    if(! $this->diafan->is_new)
    {
        echo $this->diafan->id;
    }
    else
    {
        // иначе выводим предполагаемый номер
        echo DB::query_result("SELECT MAX(id) FROM {shop_order}") + 1;
    }

    // подсказка для поля number из массива $variables
    echo $this->diafan->help();
    echo '</div>';
}
```

Хорошим тоном будет задать блоку, в котором выводим поле, идентификатор по названию поля. Это дает возможность им манипулировать. Например, скрывать/показывать если задано какое-то другое поле.

В примере мы вывели идентификатор для текущей записи (номер заказа) через переменную `$this->diafan->id`. Еще можно было бы сделать это через переменную `$this->diafan->value`. Это значение текущего поля (*number*) из базы данных. Именно в данном случае оно нам не подходит, так как в качестве номера мы используем идентификатор, а не поле *number*. А если мы хотим использовать любую другую информацию о редактируемом заказе, то получим ее через функцию `$this->diafan->values('название поля')`. То есть можно было бы и так написать: `$this->diafan->values('id')`.

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.

Пользовательская функция вызывается всегда, если она определена, независимо от типа переменной. Исключение составляет тип *module*. Теперь рассмотрим его.

Некоторые модули агрегируют информацию об элементах других модулей. При их редактировании нужно чтобы связанная информация также редактировалась.

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 таблице `$variables` нужно указать переменную, названную так же, как модуль-подключение, и задать ей тип *module*.

```
public $variables = array(
    'main' => array(
        'подключаемый_модуль' => 'module',
        ...
    ),
    ...
);
```

Пример:

При редактировании новости нужно редактировать прикрепленные изображения.

Создаем файл `modules/images/admin/images.admin.inc.php` следующего содержания:

```
class Images_admin_inc extends Diafan
{
    public function edit()
    {
        // форма добавления изображений для новости
    }
}
```

```
// ...  
}
```

В модуле новости *modules/news/admin/news.admin.php* указываем тип *module* для поля *images*. Тогда при редактировании новости подключиться вышеописанная функция *edit()*.

```
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 передана переменная `$_POST["action"] = 'save'`; и не задан модуль в переменной `$_POST["module"]`, то подключается функция `save()` из файла `adm/includes/save.php`. Функция сохраняет данные из формы на основе массива `$variables`, определенного в классе модуля.

Любую публичную функцию из файла `adm/includes/save.php` можно переопределить в файле модуля.

Пример:

При сохранении скидки нам нужно пересчитать все цены. Мы определили, что самая последняя операция при сохранении – редирект. Он выполняется функцией `save_redirect()` из файла `adm/includes/save.php`. Скопируем эту функцию в файл модуля `modules/shop/admin/shop.admin.discount.php` и добавим пересчет цен в самом начале перед редиректом. Чтобы не дублировать код, заменим его вызовом этой же функции из файла `adm/includes/save.php`.

```
// modules/shop/admin/shop.admin.discount.php
public function save_redirect()
{
    // пересчет цен
    $this->diafan->_shop->price_calc(0, $this->diafan->id);
    // вызов функции save_redirect из файла adm/includes/save.php
    parent::__call('save_redirect', array());
}
```

Также можно дополнить сохранение полей, описанных в `$variables` своей обработкой. Для этого в классе модуля определяется пользовательская функция `save_variable_переменная()`. Для файла настроек функция будет называться `save_config_variable_переменная()`.

Пользовательская функция может определять в какое поле в редактируемой таблице базы данных записать данные и как эти данные профильтровать (указываем маску). Это передаем в функцию `set_query()`:

- `$this->diafan->set_query("название_поля='%f'");`

Для определения маски обратитесь к [разделу «Базы данных»](#).

Далее с помощью функции `set_value()` задаем сами данные:

- `$this->diafan->set_value("значение_поля");`

Если данные записываются не в редактируемую таблицу, а в другую таблицу или поле вообще не принимает данные, а, например, принимает файл, то `set_query()` и `set_value()` не нужно использовать. Опишите ту логику, которая нужна.

Предыдущее значение поля (записанное в базу данных) доступно через функцию `$this->diafan->values('название поля')`.

Пример:

В модуле «Рассылки» нам нужно изменить дату рассылки только в том случае, если мы отправляем рассылку. Для того, чтобы описать это условие, мы создадим пользовательскую функцию для поля *created* (дата отправки) в файле *modules/subscription/admin/subscription.admin.php*.

```
public function save_variable_created()
{
    // значение поля оставим неизменным
    // то есть определим как уже заданное значение
    $created = $this->diafan->values("created");

    // условие: если дата не задана, то есть ранее не отправлялась рассылка
    // и в текущий момент рассылка отправляется
    if (! $created || ! empty($_POST["text"]) && ! empty($_POST["send"]))
    {
        // значение изменяем на текущее время
        $created = time();
    }

    // указываем поле, куда будут записаны данные - created
    // и маску для фильтрации - %d - числовые данные
    $this->diafan->set_query("created=%d");

    // указываем значение поля
    $this->diafan->set_value($created);
}
```

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.

Пример:

Сохранение псевдоссылки идет в отдельную таблицу базы данных *{rewrite}*. Логика работы с этой таблицей не соответствует ни одному из описанных типов данных (*text*, *numtext*, *checkbox* и т. д.). Поэтому нам нужно определить функцию для сохранения этого поля. Но в CMS уже есть такая функция в файле *adm/includes/save_functions.php*. Мы ее можем использовать. Поэтому если у нас поле называется *rewrite*, то подключиться функция *save_variable_rewrite()* из общих пользовательских функций. И нам не нужно ее описывать в файле модуля.

Пользовательская функция вызывается всегда, если она определена, независимо от типа переменной. Исключение составляет тип *module*. Теперь рассмотрим его.

Некоторые модули агрегируют информацию об элементах других модулей. При их сохранении нужно чтобы связанная информация также сохранялась.

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 таблице *\$variables* нужно указать переменную, названную так же, как модуль-подключение, и задать ей тип *module*.

```
public $variables = array(
    'main' => array(
        'подключаемый_модуль' => 'module',
        ...
    ),
    ...
);
```

Пример:

При сохранении новости нужно сохранять информацию о ссылках в меню на новость. Создаем файл *modules/menu/admin/menu.admin.inc.php* следующего содержания:

```
class Menu_admin_inc extends Diafan
{
    public function save()
    {
        // сохраняем ссылки на новость в меню в таблицу {menu}
    }
    // ...
}
```

В модуле новости *modules/news/admin/news.admin.php* указываем тип *module* для поля *menu*. Тогда при сохранении новости подключиться вышеописанная функция *save()*.

```
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 (integer \$element_id, string \$module_name, string \$element_type, [integer \$trash_id = 0]) – удаляет элементы модуля, связанные с другими элементами.

- integer \$element_id: номер удаляемого элемента
- string \$module_name: название модуля, элемент которого удаляется
- string \$element_type: тип данных
- integer \$trash_id: номер записи в корзине, с которой связано удаление

Блокирование, разблокирование

При блокировании/разблокировании 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.

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.

- `string $table`: таблица
- `array $element_ids`: номера блокируемого / разблокируемого элементов
- `integer $act`: блокировать/разблокировать

Пример:

```
/* modules/tags/admin/tags.admin.inc.php
Переопределение блокировки/разблокировки для модуля «Теги»
*/
public function act($table, $element_ids, $act)
{
    if (!$this->diafan->configmodules("tags"))
    {
        return;
    }
    if ($this->diafan->config('category'))
    {
        $table = str_replace('_category', '', $table);
        $element_ids = DB::query_fetch_value("SELECT id FROM {'.$table.'} WHERE cat_id IN
(%h)", implode(',', $element_ids), "id");
        if($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));
    }
}
```

Перемещение строки

При перемещении строки в списке подключается функции `move()` или `move_parent()` из файла `adm/includes/move.php`. Подключается та функция, которая соответствует движение – сортировка или переопределение родителя.

Обработка POST-запросов

Некоторые запросы требуют специальной обработки. Например, подгрузка списка дерева сайта при редактировании родителя страницы. Эта обработка происходит в файле *adm/includes/action.php*.

Функции для обработки общих запросов описаны в файле *adm/includes/action_functions.php*.

Если нужно производить обработку POST-запросов в модуле, то создается файл *modules/модуль/admin/модуль.admin.action.php*. В файле описывается класс *Модуль_admin_action*. Обработка данных производится в функции *action()*. 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,  
  
            // значения для списка type=select  
            'select' => array(  
                "значение1" => "отображение1",  
                "значение2" => "отображение2",  
                ...  
            )  
        )  
    )  
);
```

```

),

// значения для списка получается из базы данных
'select_db' => array(
    "table" => "название таблицы в базе данных",
    "id" => "поле, которое используется в качестве значений списка, по умолчанию id",

    // для мультязычных полей можно использовать квадратный скобки
    // пример: [text]
    "name" => "поле, которое используется для отображения пользователю в списке,
по умолчанию name",

    // пример: "trash='0' AND [act]='1'"
    "where" => "условие для SQL-запроса"

    // пример: sort ASC
    "order" => "сортировка",

    // пример: "Все"
    // если не задано, то пустое значение невозможно будет в списке выбрать
    "empty" => "пустое значение в списке",

    // значения будут выстроены с учетом вложенности
    "hierarchy" => true|false,
    ...
),

// поле выводиться, если отмечено поле, указанное в depend
// можно указать несколько полей через "," или "|"
// разделитель "," означает, что для показа поля должны быть
// отмечены все из указанных полей
// разделитель "|" означает, что для показа поля должно быть
// отмечено хотя бы одно из указанных полей
'depend' => 'влияющие_на_вывод_поле',
),
'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 атрибут *select* или *select_db*);
- 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;
- title – заголовок;
- 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 мультязычным.
 - 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. Настройки отображения:

`public $config` – настройка отображения списка значений:

- 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;
- 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);
- link_to_element – основная ссылка ведет к списку элементов, принадлежащих категории;
- multiupload – мультизагрузка изображений (подключение JS-библиотек);
- only_self – показывать только материалы редактора, если это задано в правах пользователя;
- tab_card – использование вкладок.

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);
```

4. 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 $join` – часть SQL-запроса - соединение с таблицей.

Пример:

В заказах ищем по имени пользователя, оформившего заказ.

```
$search = 'Иван';  
$this->diafan->join .= " LEFT JOIN {users} AS u ON u.id=e.user_id";  
$this->diafan->where .= " AND u.fio LIKE '%".$search."%'";
```

`public $fields` – часть SQL-запроса - дополнительные столбцы.

Пример:

Для новостей дополнительно получим значение ключевых слов.

```
$this->diafan->fields .= ", e.[keywords]";
```

5. Поля для формирования списка элементов

`public $variables_list` – массив полей для отображения в списке.

Формат:

```
public $variables_list = array (  
    'название_поля_1_латинскими_буквами' => array(  
        'type' => 'тип_поля',  
        'name' => 'название_столбца_в_шапке_списка',  
        'class' => 'CSS-класс',  
        'class_th' => 'CSS-класс_для_столбца_в_шапке_списка',  
  
        // редактирование значение прямо из списка  
        'fast_edit' => true|false,  
  
        // значение будет скрыто на узких экранах  
        'no_important' => true|false,
```

```

// значение берется из базы данных, подставляется в SQL-запрос
'sql' => true|false,

// значения для списка type=select
'select' => array(
    "значение1" => "отображение1",
    "значение2" => "отображение2",
    ...
),

// значения для списка type=select получается из базы данных
'select_db' => array(
    "table" => "название таблицы в базе данных",
    "id" => "поле, которое используется в качестве значений списка, по умолчанию id",

    // для мультязычных полей можно использовать квадратный скобки
    // пример: [text]
    "name" => "поле, которое используется для отображения пользователю, по
умолчанию name",

    // пример: "trash='0' AND [act]='1'"
    "where" => "условие для SQL-запроса",

    // пример: "Все"
    "empty" => "пустое значение в списке",
    ...
),

// любые другие атрибуты, которые могут быть использованы в пользовательских
функциях
),
'название_поля_2_латинскими_буквами' => array(
    ...
),
...
);

```

Определены следующие типы:

- function – вывод осуществляется пользовательской функцией;
- select – значения списка определяется атрибутами *select* или *select_db*. Если атрибуты не заданы, то значения списка получаются из одноименных атрибутов соответствующей переменной из массива *\$variables*;
- text – строка;
- textarea (или editor) – текст;

- numtext – число;
- floattext – число с плавающей точкой;
- none – в списке не выводиться. Обычно выводиться с атрибутом 'sql' => true и используется в пользовательской функции другой переменной.

Для любого типа можно определить пользовательскую функцию формата:

`list_variable_переменная($row, $var)`, где `$row` – массив значений для текущего элемента, `$var` – массив данных о текущей переменной.

В файле `adm/includes/show.php` описаны некоторые глобальные пользовательские функции, любую из которых можно переопределить в файле модуля. Глобальные функции `list_variable_*`() описаны для следующих переменных:

- checkbox – выводит чекбокс для групповых операций. Выводится в самом начале списка

Пример:

```
public $variables_list = array (
    'checkbox' => "",
    // ...
);
```

- sort – выводит кнопку «Перетащить» в списке элементов

Пример:

```
public $variables_list = array (
    'sort' => array(
        'name' => 'Сортировка',
        'type' => 'numtext',
        'sql' => true,
        'fast_edit' => true,
        'desc' => true,
    ),
);
```

- plus – выводит ссылку на раскрытие дерева

Пример:

```
public $variables_list = array (
    'plus' => "",
);
```

- created – выводит дату в списке

Пример:

```
public $variables_list = array (
```

```

'created' => array(
    // название столбца в шапке списка
    'name' => 'Дата и время',
    // тип может быть date или datetime
    'type' => 'datetime',
    // для отображения даты нужно ее запросить в базе данных
    'sql' => true,
    // значение будет скрыто на узком экране
    'no_important' => true,
),
);

```

- image – выводит изображение в списке

Пример:

```

public $variables_list = array (
    'image' => array(
        // название столбца в шапке списка
        'name' => 'Фото',
        // CSS-класс для столбца в шапке списка
        'class_th' => 'item__th_image ipad',
        // значение будет скрыто на узком экране
        'no_important' => true,
    ),
);

```

- name – выводит название элемента и ссылку на редактирование в списке элементов.

Пример:

```

public $variables_list = array (
    'name' => array(
        // название столбца в шапке списка
        'name' => 'Название и категория'
        // по умолчанию name
        'variable' => 'Название_поля_в_базе_данных',
        // по умолчанию не используется
        'text' => 'Текст',
    ),
);

```

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 $variables_list = array (
    'name' => array(
        'name' => 'Комментарий',
        'variable' => 'text',
    ),
);
```

Если указан атрибут *text*, то текст ссылки будет определяться этим значением.

Пример:

В модуле «Рейтинг» в основной таблице нет информации, которую можно использовать для ссылки, поэтому выводим «Редактировать». В файле *modules/rating/admin/rating.admin.php*:

```
public $variables_list = array (
    'name' => 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 $variables_list = array (
    'name' => array(
        'name' => 'Заказ',
        'variable' => 'id',
        'text' => '№ %d'
    ),
);
```

- *menu* – выводит иконку меню в списке. Функция [list_variable_menu\(\)](#) вызывается из [list_variable_name\(\)](#), но может использоваться непосредственно в списке.
- *parent* – выводит название раздела/категории в списке. Функция [list_variable_parent\(\)](#) вызывается из [list_variable_name\(\)](#) и выводит категории и разделы под названием элемента, но может использоваться непосредственно в списке.
- *adapt* – выводит верстку для адаптации под мобильный устройства в списке элементов

Пример:

```
public $variables_list = array (
    'adapt' => array(
        'class_th' => 'item__th_adapt',
    ),
);
```

- `date_period` – выводит период показа. Функция `list_variable_date_period()` вызывается из `list_variable_name()` и выводит ограничение по времени показа под названием элемента, но может использоваться непосредственно в списке.
- `actions` – выводит кнопки действий над элементом. Выводится в самом конце списка. Эта функция имеет собственные атрибуты-настройки для вывода нужных действий:
 - `act` – выводит кнопку «Сделать неактивным / Опубликовать на сайте»
 - `view` – выводит кнопку «Посмотреть на сайте»;
 - `add` – выводит кнопку «Добавить подстраницу»;
 - `del` – выводит кнопку «Удалить»;
 - `trash` – выводит кнопку «Удалить в корзину».

Пример:

```
// выведем в списке кнопки удаления в корзину и публикации на сайте
public $variables_list = array (
    // ...
    'actions' => array(
        'act' => true,
        'trash' => true,
    ),
);
```

Для удобного доступа к массиву `$variables_list` определена следующая функция:

- **`variable_list ([string $key = ''], [string $type_info = 'type'], [mixed $value = NULL])`** – возвращает, назначает информацию о поле в списке.
 - `string $key` – название переменной;
 - `string $type_info` – тип информации (`type`, `name`, `multilang`, `disabled`, `default` и др.);
 - `mixed $value` – значение. Если задано, то значение переопределяется.

Пример:

```
if ($this->diafan->variable_list('plus'))
{
    echo 'Используется иерархия элементов.';
}
if($this->diafan->variable_list('actions', 'trash'))
{
    echo 'Выводим кнопку "Удалить в корзину".';
}
```

```
// скрываем кнопку "Предпросмотр"  
$this->diafan->variable_list('actions', 'view', false);
```

6. Поля для фильтра

`public $variables_filter` – массив полей, по которым идет поиск.

Формат:

```
public $variables_filter = array (  
    'название_поля_1_латинскими_буквами' => array(  
        'type' => 'тип_поля',  
        'name' => 'название_поля/легенда',  
  
        'icon' => 'иконка_для_типа_checkbox',  
  
        // значения для типов select, radiobox, multiselect  
        'select' => array(  
            "значение1" => "отображение1",  
            "значение2" => "отображение2",  
            ...  
        ),  
  
        // значения для списка получается из базы данных  
        'select_db' => array(  
            "table" => "название таблицы в базе данных",  
            "id" => "поле, которое используется в качестве значений списка, по умолчанию id",  
  
            // для мультязычных полей можно использовать квадратный скобки  
            // пример: [text]  
            "name" => "поле, которое используется для отображения пользователю в списке, по  
            умолчанию name",  
  
            // пример: "trash='0' AND [act]='1'"  
            "where" => "условие для SQL-запроса"  
  
            // пример: sort ASC  
            "order" => "сортировка",  
  
            // значения будут выстроены с учетом вложенности  
            "hierarchy" => true|false,  
            ...  
        ),  
    ),
```

```

// выводить ссылки на временные периоды (Сегодня Месяц Год)
// для типов datetime_interval и date_interval
'links' => true|false,

// любые другие атрибуты, которые могут быть использованы в пользовательских
функциях
),
'название_поля_2_латинскими_буквами' => array(
    ...
),
...
);

```

Определены следующие типы:

- function – вывод осуществляется пользовательской функцией;
- select – выпадающий список. Значения списка определяется атрибутами *select* или *select_db*. Если атрибуты не заданы, то значения списка получаются из одноименных атрибутов соответствующей переменной из массива *\$variables* или *\$variables_list*;
- multiselect – список с выбором нескольких значений, выводиться в виде чекбоксов;
- text – поле для ввода;
- checkbox – галка-чекбокс;
- radiobox – выбор одного значения из списка-радиобокс;
- hr – горизонтальная линия;
- date – дата;
- numtext_interval – числовой интервал;
- datetime_interval – интервал даты и времени;
- date_interval – интервал даты.

Для любого типа можно определить пользовательскую функцию формата: *get_filter_variable_переменная(\$row)*, где *\$row* – массив данных о текущей переменной. Для поиска по переменным тоже можно определить пользовательскую функцию *save_filter_variable_переменная(\$row)*, где *\$row* – массив данных о текущей переменной.

В файле *adm/includes/show.php* описаны некоторые глобальные пользовательские функции, любую из которых можно переопределить в файле модуля.

Пример:

```

// функция поиска по полю "Нет изображения"
public function save_filter_variable_no_img($row)
{
    if (empty($_GET["filter_no_img"]))
    {
        return;
    }

    // добавляем условие в SQL-запрос

```



```

$this->diafan->where .= " AND (SELECT COUNT(*) FROM {images} AS i WHERE
i.element_id=e.id AND i.element_type='".$this->diafan->element_type()." AND
i.module_name='".$this->diafan->_admin->module.'" AND i.param_id=0)=0";

// учитываем заданное значение в ссылках пагинации
$this->diafan->get_nav .= ($this->diafan->get_nav ? '&' : '?' ).'filter_no_img=1';
return 1;
}

```

Для удобного доступа к массиву `$variables_filter` определена следующая функция:

- **variable_filter ([string \$key = "], [string \$type_info = 'type'], [mixed \$value = NULL])** – возвращает, назначает информацию о поле в фильтре.
 - string \$key – название переменной;
 - 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_filter('article'))
{
    echo 'В фильтре будет выведено поле для поиска по артикулу.';
}
// добавляем значения списка "Блоки форума" для фильтра
$blocks = DB::query_fetch_key_value("SELECT id, name FROM {forum_blocks} WHERE
trash='0' ORDER BY sort ASC", "id", "name");
$this->diafan->variable_filter('block_id', 'select', $blocks);

```

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. 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);
    }
}
```

9. Текущие значения

Текущий элемент для действий редактирование (*edit*), сохранение (*save*), валидация (*validate*) в переменной `$this->diafan->id`. Если элемент новый (при добавлении), то `$this->diafan->is_new = true`; При этом при редактировании и валидации `$this->diafan->id = 0`;

В пользовательских функциях `edit_variable_переменная()`, `save_variable_переменная()`, `validate_variable_переменная()` текущее значение полей доступно через функцию `values()`.

Пример:

```
echo 'Название редактируемой новости: '.$this->diafan->values('name');
```

При валидации и сохранении текущими значениями являются значения до нажатия кнопки «Сохранить».

При редактировании функция `values()` принимает два дополнительных значения:

mixed values(string \$field, [mixed \$default = "], [boolean \$save = false]) – получает значение поля.

- string \$field: название поля
- mixed \$default: значение по умолчанию, вернется, если текущее значение не определено
- boolean \$save: запомнить значение по умолчанию, вернет значение по умолчанию при дальнейшем вызове функции

Также существует переменная `$this->diafan->value`. В ней записано значение для переменной из текущей функции.

Пример:

В функции `edit_variable_created()` переменная `$this->diafan->value` будет равняться `$this->diafan->values('created')`.

10. Тип элемента

Для элементов модуля можно определить тип элемента. В стандартной редакции DIAFAN.CMS существует три типа элементов:

- `element` – стандартный тип, назначаемый элементу по умолчанию
- `cat` – категория
- `brand` – производитель

Пример:

Тип элемента используется для записи ЧПУ в таблицу `{rewrite}` для того, чтобы определить какую страницу модуля открывать – карточку товара, страницу категории или страницу производителя. Или для подключения рейтинга и к категории модуля и к элементам модуля.

Для работы с типом элемента в административной части определены следующие функции.

string **element_type()** – определяет тип элементов текущей редакции.

Пример:

```
echo 'Сейчас мы редактируем ':
switch($this->diafan->element_type())
{
    case 'cat':
        echo 'категории';
        break;

    case 'brand':
        echo 'производителей';
        break;

    case 'element':
        echo 'элементы модуля';
        break;
}
```

string **table_element_type(string \$module_name, string \$element_type)** – определяет таблицу по типу элемента.

- string `$module_name`: модуль
- string `$element_type`: тип элемента

Пример:

```
echo $this->diafan->table_element_type('shop', 'cat');  
// вернет: shop_category
```

11. Проверка текущего действия

string **is_action(\$action)** – определяет является ли текущее действие указанным в аргументе.

- string \$action: действие

Пример:

```
if($this->diafan->is_action('edit'))  
{  
    echo 'Открыта страница редактирования/добавления элемента.';  
}
```

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/admin/>.

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/](http://example.com/admin/clauses/category/)**clauses/category/**

Новые страницы заносятся в таблицы [{admin}](#).

JS-скрипты

Для работы модуля часто нужен отдельный JavaScript-сценарий. Есть несколько вариантов подключения JS-файлов. Рассмотрим их на примере.

- *modules/shop/admin/js/shop.admin.js* подключиться для редактирования товаров (на всех страницах).
- *modules/shop/admin/js/shop.admin.category.js* подключиться для редактирования категорий магазина

Еще можно добавить *edit*, тогда JS-файл подключиться только для редактирования или добавления элемента. Например, *modules/bs/admin/js/bs.admin.edit.js* подключиться только для формы редактирования/добавления баннера.

Для подключений модуля тоже автоматически подключаются:

- *modules/images/admin/js/images.admin.inc.js* – на действие *edit* при редактировании прикрепленных изображений в элементе;
- *modules/images/admin/js/images.admin.inc.config.js* – на действие *edit* при редактировании настроек изображений в модуле.

Еще автоматически подгружаются файлы для функций из файла *adm/includes/edit_functions.php*:

- *adm/js/edit/admin.edit.поле.js* – для обработки «поля»;
- *adm/js/edit/admin.edit.config.поле.js* – для обработки «поля» в настройках модуля.

Ну и также можно вручную указать:

```
$this->diafan->_admin->js_view[] = 'modules/images/admin/js/images.admin.inc.js';
```

В коде JS-файлов тоже нужно учитывать свои особенности.

Чтобы корректно отправлялся запрос, проверялся идентификационный хэш, Ajax-запросы нужно посылать через функцию *diafan_ajax.init()*:

```
diafan_ajax.init({
    data: Передаваемые_данные_в_JSON,
    url: Адрес_на_который_передаются_данные,
    success: function(response) {
        // обработка ответа response
    }
});
```

Все данные являются не обязательными. Если возвращено *response.redirect*, то происходит редирект независимо от того, что определено в *success*.

CSS-файлы

Если административная часть модуля оформляется отдельным CSS-файлом (начиная с версии 6.0.6.2), его можно подключить следующим образом:

```
$this->diafan->_admin->css_view[] = 'modules/images/css/images.admin.css';
```

Путь до файла может быть любым, в том числе абсолютным (вести на другой сайт).

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](#)
- [Обработка поисковой фразы](#)
- [Общие функции](#)
- [Парсер шаблонных тегов](#)
- [Подключение шаблонов](#)
- [Пользователи](#)
- [Почта](#)
- [Режим разработки](#)
- [Сессии](#)
- [Сжатие страниц](#)
- [Установка модулей](#)
- [Файлы](#)
- [SMS](#)

База данных

Класс с функциями для работы с базой данных – *DB* – определен в файле *includes/database.php*.

Все функции статичные, их можно вызвать из любого файла 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 и какие данные в него подставлены, запрос можно вывести на экран. Для этого нужно перед запросом написать **DEV** и включить [режим разработки](#).

Пример:

```
$text = DB::query_result("DEV SELECT [text] FROM {shop_category} WHERE id=%d", $cat_id);
```

Методы

boolean **connect** ([string \$db_url = DB_URL], [boolean \$check = false]) – Подключается к базе данных.

- string \$db_url: данные для подключения к базе данных
- boolean \$check: проверка соединения

Подключение к базе данных происходит автоматически при первом SQL-запросе. Данные для подключения берутся из константы `DB_URL`, определенной в файле `config.php`.

void **close** () – Закрывает ранее открытое соединение.

Пример:

```
// закрываем текущее соединение
DB::close();

//подключимся к другой базе данных и запросим список пользователей
DB::connect("mysql://root:@localhost/otherdb");
$users = DB::query_fetch_all("SELECT id, fio FROM otherdb_users");

// закрываем новое соединение
DB::close();

//возвращаем подключение обратно
DB::connect(DB_URL);
```

boolean **set_charset** (string \$charset) – Задаёт набор символов по умолчанию.

- string \$charset: набор символов, который необходимо установить.

Пример:

```
DB::set_charset('utf8');
```

mixed **query** (string \$query) – Отправляет запрос к базе данных.

- string \$query: текст запроса

Пример:

```
// вставляем данные в таблицу diafan_clauses
DB::query("INSERT INTO {clauses} ([name], [act], created) VALUES ('%h', '%d', %d)",
$_POST["name"], $_POST["act"], time());
```

mixed **query_without_prefix** (string \$query) – Отправляет запрос к базе данных без замены префикса.

- string \$query: текст запроса

Пример:

```
// используется при импорте базы данных, чтобы SQL-запрос не преобразовывались
DB::query_without_prefix("INSERT INTO diafan_config (name, module_name, value) VALUES
('images_variations_element', 'news', 'a:2:{i:0;a:2:
{s:4:\"name\";s:6:\"medium\";s:2:\"id\";i:1;}i:1;a:2:{s:4:\"name\";s:5:\"large\";s:2:\"id\";i:3;}})");
```

resource **query_range** (string \$query) – Отправляет запрос к базе данных с лимитом на количество получаемых в результате рядов.

- string \$query: текст запроса

Предпочтительно использовать функцию `query_range_fetch_all()`.

Пример:

```
// получаем имена трех картинок, прикрепленных к товару ID=5
$result = DB::query_range("SELECT name FROM {image} WHERE module_name='shop' AND
element_id=%d AND element_type='element'", 5, 0, 3);
```

Подобные запросы вы можете часто встретить в файлах **.model.php*, где производится выборка данных для списков, а также в модулях, поддерживающих постраничную навигацию.

Пример:

```
// запрос фотографий в альбоме ID=5 для списка
// в файле modules/photo/photo.model.php
$result = DB::query_range("SELECT * FROM {photo} WHERE cat_id=5 ORDER BY sort DESC",
$this->diafan->_paginator->polog, $this->diafan->_paginator->nastr);
```

`$this->diafan->_paginator->polog`, `$this->diafan->_paginator->nastr` в данном случае выступают в роли начального и конечного индекса выборки.

mixed **result** (resource \$result, [integer \$row = 0]) – Получает результирующие данные.

- resource \$result: обрабатываемый результат запроса
- integer \$row: номер получаемого ряда из результата

Пример:

```
// выполняем SQL-запрос к базе данных
$result = DB::query("SELECT id FROM {users} WHERE name='admin' LIMIT 1");
// получаем результат
$id = DB::result($result);
```

void **free_result** (resource \$result) – Освобождает память от результата запроса.

- resource \$result: обрабатываемый результат запроса

Пример:

```
// выполняем SQL-запрос к базе данных
$result = DB::query("SELECT id FROM {users} WHERE name='admin' LIMIT 1");

// получаем результат
$id = DB::result($result);

// освобождаем результат
DB::free_result($result);
```

В примере три операции можно заменить вызовом функции `query_result()`.

Пример:

```
$id = DB::query_result("SELECT id FROM {users} WHERE name='admin' LIMIT 1");
```

Есть аналогичные объединяющие функции для получения массива данных с одной или несколькими строками. Поэтому функции `result()` и `free_result()` вне класса `DB` не используются.

array **fetch_row (resource \$result)** – Извлекает результирующий ряд как пронумерованный массив.

- resource \$result: обрабатываемый результат запроса

Пример:

```
// запрашиваем в базе данных и выводим список товаров
$result = DB::query("SELECT id, [name] FROM {shop}");
echo 'Товары: ';
while ($row = DB::fetch_row($result))
{
    echo 'ID: '.$row[0]. ' name: '.$row[1]. "\n";
}
DB::free_result($result);
```

array **fetch_array (resource \$result)** – Извлекает результирующий ряд как массив.

- resource \$result: обрабатываемый результат запроса

Предпочтительно использовать функцию `query_fetch_array()` и аналогичные.

Пример:

```
// запрашиваем в базе данных и выводим список способов доставки
$result = DB::query("SELECT [name] FROM {shop_delivery} ORDER BY sort ASC");
echo 'Способы доставки: ';
while ($row = DB::fetch_array($result))
{
    echo $row["name"] . <br>;
}
```

```
}  
DB::free_result($result);
```

object **fetch_object (resource \$result)** – Извлекает результирующий ряд как объект.

- resource \$result: обрабатываемый результат запроса

Пример:

```
// запрашиваем в базе данных и выводим список новостей  
$result = DB::query("SELECT [name], [anons] FROM {news} ORDER BY created DESC");  
echo 'Новости:';  
while ($row = DB::fetch_object($result))  
{  
    echo 'Название: '$row->name.' анонс: '$row->anons;  
}  
DB::free_result($result);
```

integer **num_rows (resource \$result)** – Получает количество рядов в результате.

- resource \$result: обрабатываемый результат запроса

Пример:

```
// выводим количество новостей старше текущей даты  
$result = DB::query("SELECT id FROM {news} WHERE created>%d", time());  
echo DB::num_rows($result);  
DB::free_result($result);
```

integer **insert_id ()** – Возвращает автоматически генерируемый ID, используя последний запрос.

Пример:

```
// определяем ID только что добавленного пользователя  
DB::query("INSERT INTO {users} (name) VALUES ('admin')");  
$user_id = DB::insert_id();  
  
// сокращенная запись  
$user_id = DB::query("INSERT INTO {users} (name) VALUES ('admin')");
```

integer **affected_rows ()** – Возвращает число затронутых прошлой операцией рядов.

mixed **query_result ()** – Получает результирующие данные из SQL-запроса.

Пример:

```
// получаем описание текущей категории из БД  
$text = DB::query_result("SELECT [text] FROM {shop_category} WHERE id=%d", $this->diafan->cat);
```

mixed **query_fetch_array ()** – Получает результирующий ряд как массив из SQL-запроса.

Пример:

```
// запрашиваем в базе данных пользователя ID=4
$user = DB::query_fetch_array("SELECT * FROM {users} WHERE id=4");
echo 'Имя: '.$user["fio"];
echo 'Логин: '.$user["name"];
echo 'E-mail: '.$user["mail"];
```

array **query_fetch_all ()** – Получает массив результирующих рядов из SQL-запроса.

Пример:

```
// запрашиваем в базе данных всех пользователей
$users = DB::query_fetch_all("SELECT * FROM {users}");
foreach($users as $user)
{
    echo 'Имя: '.$user["fio"];
    echo 'Логин: '.$user["name"];
    echo 'E-mail: '.$user["mail"];
}
```

array **query_range_fetch_all (string \$query)** – Отправляет запрос к базе данных с лимитом на количество получаемых в результате рядов и получает массив результирующих рядов.

- string \$query: текст запроса

Пример:

```
// запрашиваем в базе данных 3 новости, начиная с 10й, не старше текущей даты
$rows = DB::query_range_fetch_all("SELECT id, [name], [anons] FROM {news} WHERE created<
%d", time(), 10, 3);
foreach($rows as $row)
{
    echo 'ID: '.$row["id"];
    echo 'Название: '.$row["name"];
    echo 'Анонс: '.$row["anons"];
}
```

array **query_fetch_key (string \$query)** –

Отправляет запрос к базе данных и получает массив результирующих рядов, в котором ключами являются значения одного из полей, название которого передано последним аргументом.

- string \$query: текст запроса

Пример:

```
// запрашиваем в базе данных новости не старше текущей даты
```

```
// результат получаем в виде массива, в котором ключами будет ID новости
$rows = DB::query_fetch_key("SELECT id, [name], [anons] FROM {news} WHERE created<%d",
time(), "id");

echo 'Новость ID=4: ';
echo 'Название: '.$rows[4]["name"];
echo 'Анонс: '.$rows[4]["anons"];

echo 'Новость ID=19: ';
echo 'Название: '.$rows[19]["name"];
echo 'Анонс: '.$rows[19]["anons"];
```

array **query_fetch_key_array (string \$query)** – Отправляет запрос к базе данных и получает массив, в котором ключами являются значения одного из полей, название которого передано последним аргументом, а значениями массив результирующих рядов, соответствующих ключу..

- string \$query: текст запроса

Пример:

```
// запрашиваем комментарии за последние 10 дней
// с группировкой по комментарию-родителю
$comments = DB::query_fetch_key_array("SELECT * FROM {comments} WHERE created>%d",
time() - 864000, "parent_id");
// комментарии первого уровня
foreach($comments[0] as $row)
{
    echo $row["text"];
    // вложенные комментарии второго уровня
    if(! empty($comments[$row["id"]]))
    {
        foreach($comments[$row["id"]] as $r)
        {
            echo $r["text"];
        }
    }
}
```

array **query_fetch_key_value (string \$query)** – Отправляет запрос к базе данных и получает массив, в котором ключами являются значения одного из полей, название которого передано предпоследним аргументом, а значениями значения другого поля, название которого передано последним аргументом..

- string \$query: текст запроса

Пример:

```
// получаем категории товаров в виде массива,
// где ключами будут ID категории, а значениями название
```



```
$cats = DB::query_fetch_key_value("SELECT id, [name] FROM {shop_category} WHERE
act='%d'", 1, "id", "name");
echo 'Название категории ID=3: '.$cats[3];
echo 'Название категории ID=5: '.$cats[5];
```

array **query_fetch_value (string \$query)** – Отправляет запрос к базе данных и получает массив значений поля, название которого передано последним аргументом..

- string \$query: текст запроса

Пример:

```
// получаем массив идентификаторов активных пользователей
$user_ids = DB::query_fetch_value("SELECT id FROM {users} WHERE act='%d'", 1, "id");
print_r($user_ids);
/* выведет:
Array
(
    [0] => 1
    [1] => 2
    [2] => 3
) */
```

mixed **query_fetch_object ()** – Получает результирующий ряд как массив из SQL-запроса.

Пример:

```
// запрашиваем в базе данных пользователя ID=4
$user = DB::query_fetch_object("SELECT * FROM {users} WHERE id=4");
echo 'Имя: '.$user->fio;
echo 'Логин: '.$user->name;
echo 'E-mail: '.$user->mail;
```

string **escape_string (string \$str)** – Мнемонизирует специальные символы в строке для использования в операторе SQL с учётом текущего набора символов/charset соединения.

- string \$str: исходная строка

Пример:

```
// подготовим данные, полученные от пользователя
// для безопасного использования непосредственно в SQL-запросе
$search = DB::escape_string($_GET["searchword"]);
DB::query("INSERT INTO {search_history} (created, name) VALUES (".time().", '".$search."");
```

Валидация данных

Методы для валидации данных объединены в класс *Validate*, описанный в файле *includes/validate.php*.

Они могут использоваться в [моделях](#) и [обработчиках POST запросов](#) непосредственно для проверки переданных данных.

Методы

string|boolean false **login (string \$name, [integer \$id = 0])** – Проверяет логин на валидность.

- string \$name: логин
- integer \$id: номер пользователя

Пример:

```
$names = array('administraor_sajta_pro_babochek', 'dbs ', 'admin', 'my_login');
foreach($names as $name)
{
    echo "'".$name."': ".Validate::login($name);
}
// выведет:
// "'': Введите логин.
// "administraor_sajta_pro_babochek": Логин больше допустимой длинны: 15.
// "dbs ": Логин не может заканчиваться пробелом.
// "admin": Пользователь с таким логином уже существует.
// "my_login":
```

string|boolean false **mail (string \$mail)** – Проверяет e-mail на валидность.

- string \$mail: электронный ящик

Пример:

```
echo Validate::mail('myemailbez@');
// выведет: E-mail адрес не правильного формата.

if(! Validate::mail('my@site.ru'))
{
    echo 'Валидация пройдена.';
}
else
{
    echo 'Валидация не пройдена.';
}
// выведет: Валидация пройдена.
```

string|boolean false **mail_user (string \$mail, [integer \$id = 0])** – Проверяет зарегистрирован ли пользователь с указанным электронным ящиком.

- string \$mail: электронный ящик
- integer \$id: номер пользователя

Пример:

```
$mes = Validate::mail_user('my@site.ru');
if($mes)
{
    echo $mes;
}
else
{
    echo 'Валидация пройдена.';
}
// выведет:
// Пользователь с таким электронным ящиком уже существует.
// или
// Валидация пройдена.
```

string|boolean false **phone (string \$phone)** – Проверяет телефон на валидность.

- string \$phone: телефон

Пример:

```
$phones = array('999-99-99', '+7 (999) 999-99-99', '999999999999');
foreach($phones as $phone)
{
    echo $phone.': ';
    $mes = Validate::phone($phone);
    if($mes)
    {
        echo $mes;
    }
    else
    {
        echo 'Валидация пройдена.';
    }
}
// выведет:
// 999-99-99: Некорректный номер.
// +7 (999) 999-99-99: Валидация пройдена.
// 999999999999: Валидация пройдена.
```

string|boolean false **password (string \$password, [boolean \$is_simple = false])** – Проверяет пароль на валидность.

- string \$password: пароль
- boolean \$is_simple: проверять сложность пароля

Пример:

```
echo Validate::password("");  
// выведет: Введите пароль.  
  
echo Validate::password('123', true);  
// выведет: Введенный пароль входит в TOP100 самых взламываемых паролей.  
Необходимо придумать другой пароль.
```

string|boolean false **numtext (string \$value)** – Проверяет число.

- string \$value: исходное значение

Пример:

```
$ints = array('345', 23, '9a');  
foreach($ints as $int)  
{  
    echo $int.': ';  
    $mes = Validate::numtext($int);  
    if($mes)  
    {  
        echo $mes;  
    }  
    else  
    {  
        echo 'Валидация пройдена.';  
    }  
}  
// выведет:  
// 345: Валидация пройдена.  
// 23: Валидация пройдена.  
// 9a: Числовое значение должно содержать только цифры.
```

string|boolean false **floattext (string \$value)** – Проверяет число с плавающей точкой.

- string \$value: исходное значение

Пример:

```
$fls = array('345', 23, '9a', '13.50', '189,45', 3456.9);  
foreach($fls as $f)  
{  
    echo $f.': ';  
    $mes = Validate::floattext($f);  
    if($mes)  
    {  
        echo $mes;  
    }  
    else  
    {
```

```

        echo 'Валидация пройдена.';
    }
}
// выведет:
// 345: Валидация пройдена.
// 23: Валидация пройдена.
// 9a: Числовое значение должно содержать только цифры и разделитель целых – точку
или запятую.
// 13.50: Валидация пройдена.
// 189,45: Валидация пройдена.
// 3456.9: Валидация пройдена.

```

string|boolean false **date (string \$value)** – Проверяет дату.

- string \$value: исходное значение

Пример:

```

$ds = array('13.05.2014', '40.05.2014', '13 мая');
foreach($ds as $d)
{
    echo $f.': ';
    $mes = Validate::date($d);
    if($mes)
    {
        echo $mes;
    }
    else
    {
        echo 'Валидация пройдена.';
    }
}
// выведет:
// 13.05.2014: Валидация пройдена.
// 40.05.2014: День не может быть больше 31.
// 13 мая: Некорректный формат даты. Введите дату в формате дд.мм.гггг.

```

string|boolean false **datetime (string \$value)** – Проверяет дату и время.

- string \$value: исходное значение

Пример:

```

$ds = array('13.05.2014', '13.05.2014 23:30', '13.05.2014 24:30', '13 мая');
foreach($ds as $d)
{
    echo $f.': ';
    $mes = Validate::datetime($d);
    if($mes)
    {
        echo $mes;
    }
}

```

```

    }
    else
    {
        echo 'Валидация пройдена.';
    }
}
// выведет:
// 13.05.2014: Валидация пройдена.
// 13.05.2014 23:30: Валидация пройдена.
// 13.05.2014 24:30: Час не может быть больше 23.
// 13 мая: Некорректный формат даты и времени. Введите дату в формате дд.мм.гггг
чч:мм.

```

`string|boolean false text (string $text, [integer $max_lenght = 40])` – Проверяет текст на наличие длинных слов.

- `string $text`: текст
- `integer $max_lenght`: максимальная длина слова

Пример:

```

echo Validate::text('В этом тексте есть
очень большое слово, которое может испортить верстку небольшой колонки.');
```

// выведет: Ошибка! Максимальная длина одного слова превышена!

`boolean url (string $url, [boolean $absolute = false])` – Проверяет URL на валидность.

- `string $url`: электронный адрес
- `boolean $absolute`: абсолютный адрес

Пример:

```

$urls = array('www.diafan.ru', 'http://www.diafan.ru/');
foreach($urls as $url)
{
    echo $url.': ';
    if(Validate::url($url, true))
    {
        echo 'Валидация пройдена.';
    }
    else
    {
        echo 'Валидация не пройдена.';
    }
}
// выведет:
// www.diafan.ru: Валидация не пройдена.
// http://www.diafan.ru/: Валидация пройдена.

```

Изображения

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.

- 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:

```
// уменьшим изображение userfiles/watermark/logo.png
// до размеров 50x100, сохраняя пропорции
Image::resize(ABSOLUTE_PATH.'userfiles/my_image.jpg', 50, 100, 90);
```

boolean `crop` (string `$original`, integer `$width`, integer `$height`, integer `$quality`, string `$vertical`, integer `$sy`, string `$horizontal`, integer `$sx`) – Обрезает изображение.

- 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

Пример:

```
// обрежем изображение userfiles/watermark/logo.png
// до размера 100x200
// отступив по 10 пикселей сверху и слева
Image::crop(ABSOLUTE_PATH.'userfiles/my_image.jpg', 1000, 200, 90, 'top', 10, 'left', 10);
```

boolean `watermark` (string `$original`, string `$watermark`, integer `$quality`, string `$vertical`, integer `$sy`, string `$horizontal`, integer `$sx`) – Добавляет водяной знак на изображение.

- 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 вертикале
- string \$horizontal: горизонтальное расположение знака (left, center, right)
- integer \$sx: отступ по горизонтали

Пример:

```
// наложим водяной знак userfiles/watermark/logo.png
// на изображение userfiles/my_image.jpg,
// отступив по 10 пикселей справа и внизу
Image::watermark(ABSOLUTE_PATH.'userfiles/my_image.jpg',
ABSOLUTE_PATH.'userfiles/watermark/logo.png', 90, 'bottom', 10, 'right', 10);
```

boolean **wb (string \$original, integer \$quality)** – Обесцвечивает изображение.

- string \$original: путь к файлу
- integer \$quality: качество изображения

Пример:

```
// применим к изображению userfiles/my_image.jpg фильтр
// IMG_FILTER_GRAYSCALE, переводящий его в оттенки серого
Image::wb(ABSOLUTE_PATH.'userfiles/my_image.jpg', 90);
```


Кастомизация

- [Частичная кастомизация](#)

Класс для внедрения пользовательских разработок – *Custom* – определен в файле *includes/custom.php*.

Этот класс позволяет подключить файл из текущей темы, если он там есть. Кроме того, класс обеспечивает частичную кастомизацию, которая позволяет веб-мастеру выносить свои доработки из файла, который он изменяет.

Методы

void **init** ([string \$name = '']) – Обнуляет внутренний кэш класса.

- string \$name

Пример:

```
// инициация класса Custom сразу после подключения
include_once ABSOLUTE_PATH.'includes/custom.php';
Custom::init();
```

string **add** (string \$name) – Добавляет название примененной темы.

- string \$name

string **del** ([string \$name = '']) – Исключает название примененной темы.

- string \$name

array **names** () – Возвращает названия примененных тем.

void **inc** (string \$path) – Подключает PHP-файл.

- string \$path: путь до файла относительно корня сайта

Пример:

```
// подключаем файл includes/sms.php или файл
// custom/my_theme/includes/sms.php, если файл существует
// и тема my_theme является текущей
Custom::inc('includes/sms.php');
```

boolean **exists** (string \$path_to_file) – Проверяет существует ли файл.

- string \$path_to_file: путь до файла относительно корня сайта

Пример:

```
if(Custom::exists('includes/sms.php'))
{
    echo 'ok';
}
```

```

}
else
{
    echo 'fail';
}
// выведет ok, если файла есть в основном хранилище скриптов
// или в текущей теме

```

string **path (string \$path_to_file)** – Возвращает путь до файла.

- string \$path_to_file: путь до файла относительно корня сайта

Пример:

```

echo Custom::path('includes/sms.php');
// выведет:
// custom/my_theme/includes/sms.php, если такой файл существует
// и тема my_theme является текущей
// или includes/sms.php в ином случае

// заменяем Custom::inc('includes/sms.php');
// если нужно, чтобы переменные, определенные файл были доступны
// в контексте файла, в котором идет подключение
include_once(Custom::path('includes/sms.php'));

```

array **read_dir (string \$path)** – Читает директорию.

- string \$path: путь до директории относительно корня сайта

Пример:

```

// читает папку modules в основном хранилище файлов и добавляет
// файлы и директории из текущей темы
$rows = Custom::read_dir('modules');
print_r($rows);
/* выведет:
Array
(
    [0] => ab
    [1] => admin
    [2] => attachments
    ...
    [45] => wishlist
)
*/

```

string **version_core ()** – Возвращает версию сборки.

Частичная кастомизация

Частичная кастомизация позволяет выносить доработки из файла, который изменяется.

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 файле частичной кастомизации можно переопределить классы, свойства и функции классов. Также можно 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 `modules/users/users.inc.php`. Для этого создаем файл `modules/users/users.inc.custom.php`.

следующего содержания:

```
class Users_inc extends Model
{
    // 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.

Если на сайте используется несколько примененных тем, то порядок подключения файлов частичной кастомизации соответствует дате подключения тем. То есть сначала на основной файл накладывается файл частичной кастомизации из первой примененной темы. Затем на сгенерированный файл накладывается файл частичной кастомизации из второй примененной темы. Таким образом при приставки *after*, *before* могут быть использованы в двух темах и код будет учитываться из обеих тем. А приставка *replace* из последней примененной темы заменит код, перекрыв изменения из предыдущих тем.

Конфигурация сайта

Работа с файлом *config.php* осуществляется с помощью подключения «Конфигурация сайта» (файл *includes/config.php*, класс *Config*).

Методы

void save (array \$new_values, array \$languages, [string \$save_demo = false]) – Сохраняет параметры сайта.

- array \$new_values: новые значения параметров сайта
- array \$languages: языки сайта
- string \$save_demo: сохранить файл конфигурации для демо-версии

Пример:

```
Custom::inc('includes/config.php');

//изменяем параметр «Режим разработки»
$new_values = array('MOD_DEVELOPER' => true);
$languages = array(
    array('id' => 1),
    array('id' => 2)
);
Config::save($new_values, $languages);
```

Кэширование

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

void close () – Закрывает ранее открытое соединение.

mixed get (string|array \$name, string \$module, [boolean \$developer = false]) – Читает кэш модуля \$module с меткой \$name.

- string|array \$name: метка кэша
- string \$module: название модуля
- boolean \$developer: флаг принудительно записи в кэш

В метке кэша нужно передать все условия, при которых содержимое может меняться (номер страницы, язык сайта и пр.).

Пример:

```
// запрашиваем данные из кэша для первой страницы модуля Вопрос-Ответ
$cache_meta = array(
    // уникальное название
    "name" => "first_page",
    // язык сайта
    "lang_id" => _LANG,
    // страница сайта, к которой прикреплен модуль
    "site_id" => $this->diafan->_site->id,
    // роль пользователя
    "role_id" => $this->diafan->_users->role_id ? $this->diafan->_users->role_id : 0,
    // время
    "time" => $time
);
$result = $this->diafan->_cache->get($cache_meta, "faq");
```

boolean save (mixed \$data, string|array \$name, string \$module, [boolean \$developer = false]) – Сохраняет данные \$data для модуля \$module с меткой \$name.

- mixed \$data: данные, сохраняемые в кэше
- string|array \$name: метка кэша
- string \$module: название модуля
- boolean \$developer: флаг принудительно записи в кэш

Пример:

```
$this->diafan->_cache->save($result, $cache_meta, "faq");
```

Методы `get()` и `save()` используются вместе.

Пример:

```
$cache_meta = array(
    "name"    => "first_page",
    "lang_id" => _LANG,
    "site_id" => $this->diafan->_site->id,
    "role_id" => $this->diafan->_users->role_id ? $this->diafan->_users->role_id : 0,
    "time"    => $time
);
if(! $result = $this->diafan->_cache->get($cache_meta, "faq"))
{
    // генерируем данные и записываем их в переменную $result

    //сохранение кеша
    $this->diafan->_cache->save($result, $cache_meta, "faq");
}
```

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.

- string|array \$name: метка кэша
- string \$module: название модуля

Пример:

```
// удаляем кэш модуля Интернет-магазин
$this->diafan->_cache->delete("", "shop");

// удаляем кэш всех модулей
$this->diafan->_cache->delete("", array());
```

Маршрутизатор

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 POST-запросов](#).

Свойства

Свойства класса *Route* – переменные передаваемые в URL. Список всех доступных переменных задан в файле *adm/includes/init.php* (для административной части) и *includes/init.php* (для пользовательской части) и доступен через массив `$this->diafan->rewrite_variable_names_all`.

Переменные, передаваемые в URL в пользовательской части:

integer **page** – номер страницы для пагинации.

Пример:

```
// http://site.ru/news/page2/  
// вторая страница списка новостей, если отключены категории  
$this->diafan->_route->page = 2;
```

integer **show** – идентификатор элемента модуля, если отключено автоматическое формирование ЧПУ.

Пример:

```
// http://site.ru/news/show2/  
// страница новости ID=2  
$this->diafan->_route->show = 2;
```

integer **add** – добавление нового элемента.

Пример:

```
// http://site.ru/ads/add1/  
// страница добавления объявления  
$this->diafan->_route->add = 1;
```

integer **edit** – редактирование элемента.

Пример:

```
// http://site.ru/ads/edit15/  
// страница редактирования объявления ID=15  
$this->diafan->_route->edit = 15;
```

integer **param** – идентификатор значения дополнительной характеристики.

Пример:

```
// http://site.ru/shop/param3/  
// страница товаров красного (ID=3) цвета  
$this->diafan->_route->param = 3;
```

integer **dpage** – номер страницы для дополнительной пагинации.

Пример:

```
// http://site.ru/shop/palatki/52d/dpage3/  
// третья страница списка комментариев на странице товара  
$this->diafan->_route->dpage = 3;
```

integer **cat** – идентификатор категории.

Пример:

```
// http://site.ru/shop/cat5/  
// страница категории товаров ID=5  
$this->diafan->_route->cat = 5;
```

integer **brand** – идентификатор производителя.

Пример:

```
// http://site.ru/shop/brand1/  
// страница производителя товаров ID=1  
$this->diafan->_route->brand = 1;
```

integer **year** – год.

Пример:

```
// http://site.ru/news/year2013/  
// список новостей за 2013 год  
$this->diafan->_route->year = 2013;
```

integer **month** – месяц.

Пример:

```
// http://site.ru/ads/year2013/month11/  
// список объявлений за ноябрь 2013 года  
$this->diafan->_route->month = 11;
```

integer **step** – шаг.

Пример:

```
// http://site.ru/shop/cart/step2/  
// шаг 2 при оформлении заказа в магазине  
$this->diafan->_route->step = 2;
```

integer **sort** – направление сортировки.

Пример:

```
// http://site.ru/shop/palatki/sort2/  
// список товаров в категории с сортировкой по убыванию цены  
$this->diafan->_route->sort = 2;
```

Переменные, передаваемые в URL в административной части:

integer **page** – номер страницы для пагинации.

Пример:

```
// http://site.ru/admin/news/page2/  
// вторая страница списка новостей  
$this->diafan->_route->page = 2;
```

integer **addnew** – добавление нового элемента.

Пример:

```
// http://site.ru/admin/ab/category/addnew1/  
// страница добавления категории объявлений  
$this->diafan->_route->addnew = 1;
```

integer **edit** – редактирование элемента.

Пример:

```
// http://site.ru/admin/news/edit15/  
// страница редактирования новости ID=15  
$this->diafan->_route->edit = 15;
```

integer **site** – идентификатор раздела сайта.

Пример:

```
// http://site.ru/admin/shop/site3/  
// список товаров, прикрепленных к странице сайта ID=3  
$this->diafan->_route->site = 3;
```

integer **cat** – идентификатор категории.

Пример:

```
// http://site.ru/admin/shop/cat5/  
// список товаров, прикрепленных к категории ID=5  
$this->diafan->_route->cat = 5;
```

integer **parent** – идентификатор родителя.

Пример:

```
// http://site.ru/admin/site/parent4/  
// список страниц сайта, раскрыты вложенные страницы для пункта ID=4  
$this->diafan->_route->parent = 4;
```

integer **error** – номер ошибки.

Пример:

```
// http://site.ru/admin/shop/error7/  
// страница, на которой будет выведена ошибка:  
// Внимание! Не установлена библиотека GD. Работа модуля невозможна. Обратитесь в  
// техподдержку вашего хостинга!  
$this->diafan->_route->error = 7;
```

integer **success** – сообщения об успешном выполнении.

Пример:

```
// http://site.ru/admin/shop/success1/  
// страница, на которой будет выведено сообщение:  
// Изменения сохранены!  
$this->diafan->_route->success = 1;
```

Свойства

var **variable_names_site** = array('cat', 'param', 'show', 'dpage', 'rpage', 'brand', 'year', 'month', 'day', 'step', 'sort', 'add', 'edit', 'page') – переменные, передаваемые в URL в пользовательской части.

var **variable_names_admin** = array('edit', 'savenew', 'save', 'addnew', 'site', 'cat', 'parent', 'page', 'step', 'error', 'success') – переменные, передаваемые в URL в административной части.

Методы

void **save** (string \$rewrite, string \$text, integer \$element_id, string \$module_name, string \$element_type, integer \$site_id, [integer \$cat_id = 0], [integer \$parent_id = 0], [boolean \$add_parents = false], [boolean \$change_children = false]) – Сохраняет ЧПУ.

- string \$rewrite: псевдоссылка
- string \$text: значение, из которого автоматически генерируется ЧПУ
- integer \$element_id: номер элемента модуля
- string \$module_name: модуль

- string \$element_type: тип данных (*element* – элемент (по умолчанию), *cat* – категория, *param* – значение списка доп.характеристики, *brand* – производитель)
- integer \$site_id: номер страницы сайта
- integer \$cat_id: номер категории
- integer \$parent_id: номер родителя
- boolean \$add_parents: добавлять ЧПУ родительских элементов
- boolean \$change_children: изменить ЧПУ у детей

Пример:

```
// генерируем и сохраняем ЧПУ для товара ID=54 из имени, согласно настройкам
// в начале ЧПУ будет добавлено ЧПУ категории ID=2
// или страницы сайта ID=3
$this->diafan->_route->save('','Рюкзак «Контур 50»', 54, 'shop', 'element', 3, 2, 0, true);
```

```
// сохраняем ЧПУ для страницы сайта ID=5, заменяем первую часть ЧПУ
// у вложенных страниц
$this->diafan->_route->save('about', 'О компании', 5, 'site', 'element', 5, 0, 0, true, true);
/* к примеру была следующая структура страниц:
О компании /company/
  История /company/history/
  Философия /company/philosophy/
  Обратная связь /feedback/
```

```
результат работы функции:
О компании /about/
  История /about/history/
  Философия /about/philosophy/
  Обратная связь /feedback/
*/
```

void **delete** (integer|array \$element_ids, string \$module_name, [string \$element_type = 'element']) – Удаляет ЧПУ одного или нескольких элементов.

- integer|array \$element_ids: номер одного или нескольких элементов
- string \$module_name: название модуля
- string \$element_type: тип данных

string **generate_rewrite** (string \$text) – Генерирует псевдоссылку.

- string \$text: исходный текст

Пример:

```
$rewrite = $this->diafan->_route->generate_rewrite('Рюкзак «Контур 50»');
// вернет: ryukzak-kontur-50, если метод преобразования в «Параметрах сайта» задан
как «транслит»
```

string **link** (integer \$site_id, [integer \$element_id = 0], [string \$module_name = ''], [string \$element_type = 'element'], [boolean \$insert_route_end = true]) – Генерирует ссылку.

- integer \$site_id: номер страницы сайта
- integer \$element_id: номер элемента
- string \$module_name: модуль
- string \$element_type: тип данных (*element* – элемент (по умолчанию), *cat* – категория, *param* – значение списка доп.характеристики, *brand* – производитель)
- boolean \$insert_route_end: добавлять окончание ЧПУ в конце ссылки

Пример:

```
// получим ссылку на страницу производителя ID=1,
// прикрепленного к странице сайта ID=3
$link = $this->diafan->_route->link(3, 1, "shop", "brand");

// получим ссылку на страницу сайта ID=5
$link = $this->diafan->_route->link(5);
```

void **prepare (integer \$site_id, integer \$element_id, string \$module_name, [string \$element_type = 'element'])** – Подготавливает ЧПУ.

- integer \$site_id: номер страницы сайта
- integer \$element_id: номер элемента
- string \$module_name: модуль
- string \$element_type: тип данных (*element* – элемент (по умолчанию), *cat* – категория, *param* – значение списка доп.характеристики, *brand* – производитель)

Пример:

```
// в этом примере будет выполнено три SQL-запроса к базе данных
// для получения ЧПУ всех заданных товаров
$sids = array(3, 5, 7);
foreach($sids as $sid)
{
    $link[$sid] = $this->diafan->_route->link(3, $sid, "shop");
}
```

Пример:

```
// в этом примере будет выполнен один SQL-запрос к базе данных
// для получения ЧПУ всех заданных товаров
$sids = array(3, 5, 7);
foreach($sids as $sid)
{
    $this->diafan->_route->prepare(3, $sid, 'shop');
}
foreach($sids as $sid)
{
    $link[$sid] = $this->diafan->_route->link(3, $sid, "shop");
}
```

string|boolean false **module (string \$module_name, [boolean \$route_end = true])** – Получает ЧПУ страницы сайта по названию модуля.

- string \$module_name: название модуля
- boolean \$route_end: выводить окончание

Пример:

```
// получаем ссылку на страниц корзины
$link = BASE_PATH_HREF.$this->diafan->_route->module("cart");
```

mixed **id_module (string \$module_name, [mixed \$site_id = 0], [boolean \$return_array = true])** – Определяет номер страницы, к которой прикреплен модуль, доступной текущему пользователю.

- 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:

```
// определяем ID страницы сайта, к которой подключен модуль «Личные сообщения»
$site_id = $this->diafan->_route->id_module("messages", 0, false);

// из переменных ID выбираем те, к которым подключен модуль «Магазин»
$site_ids = $this->diafan->_route->id_module("shop", array(3, 5, 6));
```

string **current_link ([string|array \$exclude = "], [array \$include = array()])** – Выдает URL текущей страницы с включенными или исключенными переменными.

- string|array \$exclude: исключенные переменные
- array \$include: включенные переменные

Пример:

```
// убираем из адреса текущей страницы переменные page и dpage и добавляем add1/
$link = $this->diafan->_route->current_link(array("page", "dpage"), array("add" => 1));
// например, для страницы http://site.ru/shop/palatki/page3/dpage2/sort2/
// результат будет http://site.ru/shop/palatki/sort2/add1/
```

string **current_admin_link ([string|array \$exclude = "])** – Выдает URL текущей страницы административной части с включенными переменными.

- string|array \$exclude: исключенные переменные

Пример:

```
// убираем из адреса текущей страницы переменные page и parent
$link = $this->diafan->_route->current_admin_link(array("page", "parent"));
```

```
// например, для страницы http://site.ru/admin/site/parent15/page2/  
// результат будет http://site.ru/admin/site/
```

string **replace_link_to_id (string \$text)** – Заменяет ссылки на идентификаторы.

- string \$text: исходный текст

Пример:

```
// обрабатываем текст, добавленный через визуальный редактор  
// перед сохранением в базу данных  
$text = $this->diafan->_route->replace_link_to_id($_POST["text"]);
```

string **replace_id_to_link (string \$text)** – Заменяет идентификаторы ссылки на ЧПУ.

- string \$text: исходный текст

Пример:

```
// преобразуем текст, полученных из базы данных перед выводом на экран  
echo $this->diafan->_route->replace_id_to_link($text);
```

Обработка поисковой фразы

Фраза для поиска подготавливается специальным образом для поиска по базе. За это отвечает класс *Searchwords* в файле *includes/searchwords.php*.

Свойства

var **max_length** – максимальная длина слова.

Методы

array **prepare (string \$text)** – Выделение уникальных слов из текста.

- string \$text: индексируемый текст

Функция использует [стеммер Потера](#).

Пример:

```
Custom::inc('includes/searchwords.php');
$searchwords = new Searchwords();
$array = $searchwords->prepare('Не только стильный, но и достаточно удобный в
эксплуатации. Его основной объем и два больших боковых кармана на молнии, вместят
все необходимое для рыбной ловли или охоты.');
```

print_r(\$array);

```
/* выведет:
Array
(
    [0] => только
    [1] => стильн
    [2] => достаточно
    [3] => удобн
    [4] => эксплуатац
    [5] => его
    [6] => основн
    [7] => объ
    [8] => дв
    [9] => больших
    [10] => боковых
    [11] => карма
    [12] => молн
    [13] => вместят
    [14] => вс
    [15] => необходим
    [16] => дл
    [17] => рыбн
    [18] => ловл
    [19] => ил
    [20] => охот
)
*/
```


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:

```
$this->diafan->configmodules("list_img");
```

Методы

void redirect ([string \$url = "], [integer \$http_response_code = 302]) – Редирект.

- string \$url: URL для редиректа
- integer \$http_response_code: статус-код

Пример:

```
// редирект на страницу новостей
$this->diafan->redirect('news/');

// редирект на внешний сайт
$this->diafan->redirect('http://www.diafan.ru/');

// редирект в административной части
// на редактирование категорий новостей
$this->diafan->redirect('news/category/');
```

void redirect_js ([string \$url = "], [boolean \$no_history = false]) – Редиректс помощью JavaScript.

- string \$url: URL для редиректа
- boolean \$no_history: не сохранять исходную страницу в истории сеансов

Используется в тех случаях, когда на экран уже выведен контент. Тогда редирект с помощью функции `redirect()` не работает.

Пример:

```
// редирект на главную страницу сайта
$this->diafan->redirect_js("");
```

mixed filter (mixed \$array, [string \$mask = 0], [string \$name = "], [mixed \$default = "]) – Приводит значение переменной к типу, соответствующему маске.

- mixed \$array: исходное значение или массив с исходным значением
- string \$mask: тип преобразования: *url* – преобразует строку для использования ее в ссылке, *sql* – переменную можно вставлять непосредственно в SQL-запрос, *int* –

оставляет только числа, *float* – дискретное число, *string* – удаляются HTML-теги, специальные символы преобразуются

- `string $name`: имя переменной в массиве
- `mixed $default`: значение по-умолчанию

Пример:

```
// получает числовое значение из переменной $_POST["number"]
$integer = $this->diafan->filter($_POST, 'int', "number");

// преобразует значение переменной $_GET["searchword"]
// для того, чтобы безопасно использовать его для составления ссылок
$search = $this->diafan->filter($_GET, 'url', "searchword");
$url = BASE_PATH.'search/?searchword='.$search;

// убираем HTML-теги и преобразуем спец.символы в переменной $anons
$anons = $this->diafan->filter($anons, 'string');

// преобразуем значение переменной $name для того,
// чтобы подставить непосредственно в SQL-запрос
$user = DB::query_fetch_array("SELECT * FROM {users} WHERE name='".$this->diafan->filter($name, 'sql')."'");
```

`mixed configmodules (string $name, [string $module_name = ''], [integer $site_id = false], [integer $lang_id = false], [boolean $value = false])` –

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.

- `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:

```
if($this->diafan->configmodules("cat", "shop"))
{
    echo 'В модуле «Магазин» подключены категории.'
}
```

`string short_text (string $text, [integer $length = 80])` – Сокращает текст.

- `string $text`: исходный текст
- `integer $length`: количество символов для сокращения

Пример:

```
echo $this->diafan->short_text('<p>Не только стильный, но и достаточно удобный в эксплуатации. Возможна навеска дополнительного снаряжения снаружи. </p>');  
// вернет:  
// Не только стильный, но и достаточно удобный в эксплуатации. Возможна навеска дополнительного...  
  
echo $this->diafan->short_text('<p>Не только стильный, но и достаточно удобный в эксплуатации. Возможна навеска дополнительного снаряжения снаружи. </p>', 400);  
// вернет:  
// Не только стильный, но и достаточно удобный в эксплуатации. Возможна навеска дополнительного снаряжения снаружи.
```

string **prepare_xml (string \$text)** – Подготавливает текст для отображения в XML-файле.

- string \$text: исходный текст

Пример:

```
// обработаем название новости для корректного  
// формирования RSS-ленты новостей  
echo '<title>'. $this->diafan->prepare_xml($row['name']). '</title>';
```

string **convert (integer \$size)** – Конвертирует количество бит в байты, килобайты, мегабайты.

- integer \$size: размер в байтах

Пример:

```
echo $this->diafan->convert(45980);  
// вернет: 44.9 Kb  
  
echo $this->diafan->convert(367800342);  
// вернет: 350.76 Mb
```

string **encrypt (string \$text)** – Кодировать пароль.

- string \$text: исходный пароль

Пример:

```
echo $this->diafan->encrypt('123');  
// вернет: 944104956716c66f4c36a236ae349053
```

array **get_children (integer \$id, string \$table, [boolean \$trash = true])** – Выдает массив номеров детей.

- integer \$id: номер исходного элемента
- string \$table: таблица
- boolean \$trash: не учитывать элементы, удаленные в корзину

Пример:

```
// получаем ID всех вложенных пунктов меню для пункта меню ID=3,  
// не удаленных в корзину  
$children = $this->diafan->get_children(3, "menu", true);
```

array **get_parents (integer|array \$id, string \$table)** – Выдает массив номеров родителей.

- integer|array \$id: номер исходного элемента
- string \$table: таблица

Пример:

```
// получаем ID всех родителей для текущей страницы сайта  
$parents = $this->diafan->get_parents($this->diafan->_site->id, 'site');
```

string **translit (string \$text)** – Переводит кириллицу в транслит для строки text.

- string \$text: исходный текст

Пример:

```
echo $this->diafan->translit('Рюкзак «Контур 50»');  
// выведет: Ryukzak-Kontur-50
```

string **formate_from_date (string \$date)** – Переводит дату из в формате гggг-мм-дд в формат дд.мм.гггг.

- string \$date: дата в формате гggг-мм-дд

Пример:

```
echo $this->diafan->formate_from_date('2014-05-13');  
// выведет: 13.05.2014
```

string **formate_from_datetime (string \$date)** – Переводит дату из в формате гggг-мм-дд чч:мм в формат дд.мм.гггг чч:мм.

- string \$date: дата в формате гggг-мм-дд чч:мм

Пример:

```
echo $this->diafan->formate_from_datetime('2014-05-13 18:30');  
// выведет: 13.05.2014 18:30
```

string **formate_in_date (string \$date)** – Переводит дату из в формате дд.мм.гггг в формат гggг-мм-дд.

- string \$date: дата в формате дд.мм.гггг

Пример:

```
echo $this->diafan->formate_in_date('13.05.2014');  
// выведет: 2014-05-13
```

string **formate_in_datetime (string \$date)** – Переводит дату из в формата дд.мм.гггг чч:мм в формат гггг-мм-дд чч:мм.

- string \$date: дата в формате дд.мм.гггг чч:мм

Пример:

```
echo $this->diafan->formate_in_datetime('13.05.2014 18:30');  
// выведет: 2014-05-13 18:30
```

integer **unixdate (string \$date)** – Возвращает дату, переданную в формате dd.mm.yyyy hh:ii в виде даты в формате UNIX.

- string \$date: дата в формате dd.mm.yyyy hh:ii

Пример:

```
echo $this->diafan->unixdate('13.05.2014 18:30');  
// выведет: 1399991400
```

string **version_core ()** – Возвращает версию сборки.

string **uid ([boolean \$flag = false])** – Генерирует уникальный идентификатор.

- boolean \$flag: версия идентификатора без сокращения

string **domain ([boolean \$without_mobile = false])** – Возвращает доменное имя.

- boolean \$without_mobile: без указания мобильной версии

string **get_http_status (string \$url)** – Возвращает HTTP статус ответа сервера.

- string \$url: URL-адрес

mixed(boolean|string|array) **fast_request (string \$url, [array \$params = false], [array \$method = "GET"], [boolean \$answer = false], [boolean \$debug = false])** – Инициализация быстрого запроса.

- string \$url: URL-адрес
- array \$params: параметры запроса
- array \$method: метод передачи параметров: "GET", "POST", "POST_JSON" - метод POST передачи JSON-представления данных
- boolean \$answer: вернуть ответ сервера
- boolean \$debug: вернуть заголовки запроса и ответа

string **params_append (string \$url, array|object \$params)** – Appends the parameters of the object/array \$params to the main URL.

- string \$url
- array|object \$params

mixed **simple_request** (**string** \$url, [**array** \$params = false], [**string** \$filename = false], [**string** \$content = false]) – Инициализация простого запроса.

- string \$url: URL-адрес
- array \$params: параметры запроса
- string \$filename: путь до файла относительно корня сайта или url-адрес
- string \$content: содержание файла / контекст запроса

array **array_column** ([**array** \$input = null], [**mixed** \$columnKey = null], [**mixed** \$indexKey = null]) –

Returns the values from a single column of the input array, identified by the \$columnKey.

Optionally, you may provide an \$indexKey to index the values in the returned array by the values from the \$indexKey column in the input array.

- array \$input: A multi-dimensional array (record set) from which to pull a column of values.
- mixed \$columnKey: The column of values to return. This value may be the integer key of the column you wish to retrieve, or it may be the string key name for an associative array.
- mixed \$indexKey: (Optional.) The column to use as the index/keys for the returned array. This value may be the integer key of the column, or it may be the string key name.

array **getcsv** (**string** \$st, [**string** \$d = ";"], [**string** \$q = "'']) – Производит разбор данных CSV.

- string \$st: строка
- string \$d: символ разделителя поля
- string \$q: символ ограничителя поля

Парсер шаблонных тегов

Шаблонные теги вызывают исполнение функций или подключают файлы. Разбор файлов шаблона и логика подключения описана в файле *includes/parser_theme.php*, класс *Parser_theme*. В модуле объект этого класса доступен через переменную `$this->diafan->_parser_theme`, в общих шаблонных тегах (файлы из папки **themes/functions**) объект класса доступен через переменную `$this`.

Свойства

var **current_module** – текущий модуль, для которого вызвана шаблонная функция.

var **is_tag** – в текущий момент исполняется шаблонный тег.

Методы

void **show_theme ()** – Подключает шаблон.

Пример:

```
// подключает шаблон страницы в файле includes/init.php
$this->_parser_theme->show_theme();
```

string **get_function_in_theme (string \$text, [boolean \$php = false])** – Парсит шаблон.

- string \$text: содержание шаблона
- boolean \$php: исполнять PHP-код

Пример:

```
$text = 'Здесь выводиться блок новостей: <insert name="show_block" module="news">';
echo $this->diafan->_parser_theme->get_function_in_theme($text);
/* Результат:

    Здесь выводиться блок новостей:

    1 мая
    Начало майских праздников

    2 мая
    Продолжение майских праздников
*/
```

void **functions (string \$name, [array \$attributes = array()])** – Подключает файл, описывающий общий шаблонный тег.

- string \$name: название шаблонного тега
- array \$attributes: атрибуты шаблонного тега

Пример:

```
// подключаем шаблонный тег <insert name="show_h1">
// в теге <insert name="show_body">
// файл themes/functions/show_body.php
$this->functions('show_h1');

// подключаем шаблонный тег <insert name="show_year" year="2009">
$this->functions('show_year', array('year' => 2009));
```

array **get_attributes (array \$attributes)** – Задаёт неопределённым атрибутам шаблонного тега значение по умолчанию.

- array \$attributes: массив определённых атрибутов

Пример:

```
// задает атрибуты для тега <insert name="show_href">
// файл themes/functions/show_href.php
$attributes = $this->get_attributes($attributes, 'rewrite', 'img', 'img_act', 'width', 'height', 'alt');
```


Подключение шаблонов

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

string get (string \$name, string \$module, mixed \$result, [string \$template = "]) – Подключает шаблон.

- string \$name: имя функции
- string \$module: название модуля
- mixed \$result: передаваемые в шаблон данные
- string \$template: атрибут template в шаблонном теге

Пример:

```
// подключаем шаблон формы блока объявлений если существует шаблон
// modules/ab/views/ab.view.show_block_{$attributes["template"]}.php
// то подключаем его,
// иначе подключаем шаблон modules/ab/views/ab.view.show_block.php
echo $this->diafan->_tpl->get('show_block', 'ab', $result, $attributes["template"]);
```

void js ([string \$name = "], [string \$module = "]) – Подключает JS-файл.

- string \$name: часть имени файла
- string \$module: название модуля

Пример:

```
// includes/controller.php
// подключаем JS-файл, общий для всего модуля
$this->diafan->_tpl->js();
```

void css ([string \$name = "], [string \$module = "]) – Подключает CSS-файл.

- string \$name: часть имени файла
- string \$module: название модуля

string htmleditor (string \$text) – Заменяет шаблонные теги, ссылки в тексте.

- string \$text: исходный текст

Пример:

```
// выводим шаблонным тегом динамический блок
// в шаблоне modules/news/views/news.view.id.php
echo $this->htmleditor('<insert name="show_dynamic" module="site" id="1">');

// обрабатываем текст новости, редактируемый через визуальный редактор
```

```
// в файле modules/news/news.model.php  
$row["text"] = $this->diafan->_tpl->htmleditor($row["text"]);
```

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.action.php`).

Методы

boolean **send_mail** (string|array \$recipient, string \$subject, string \$body, [string \$from = "], [string &\$error_output = "], [string &\$trace_output = "]) – Отправляет электронное письмо.

- string|array \$recipient: получатель/получатели
- string \$subject: тема письма
- string \$body: содержание письма
- string \$from: адрес отправителя
- string \$error_output: вывод ошибки
- string \$trace_output: вывод трассировки

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">
```

```






```

В файле *modules/shop/views/shop.view.id.php* в напомним:

```
$this->get('email_form', 'shop', $result);
```

В контроллер добавляем *modules/shop/shop.php*

```

switch ($_POST['action'])
{
    case 'mail':
        return $this->action->send_friends();
    // ...
}

```

Теперь обработчик POST-запросов *modules/shop/shop.action.php*

```

public function send_friends()
{
    if (empty($_POST['mail']))
    {
        $this->result['errors']['mail'] = 'Пожалуйста, введите e-mail.';
    }
    else
    {
        $this->valid_email($_POST['mail'], "mail");
    }
    $row = DB::query_fetch_array("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.';
    }
}

```

```
}

if ($this->result())
    return;

$row['link'] = BASE_PATH_HREF.$this->diafan->_route->link($row["site_id"], "shop",
$row["id"]);

Custom::inc(ABSOLUTE_PATH.'includes/mail.php');
$subject = "Рекомендую посмотреть";
$message = 'Здравствуйте! Ваш друг рекомендует Вам посмотреть <a href="'.
$row['link'].>'.</a>';
$from_mail = $this->diafan->configmodules("emailconf", 'shop') ? $this->diafan-
>configmodules("email", 'shop') : "";

$this->result['errors'][0] = send_mail($_POST['mail'], $subject, $message, $from_mail) ?
'Отправлено' : 'Ошибка';
}
```

Режим разработки

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

void **init ()** – Разрешает/запрещает вывод ошибок.

boolean **set_profiling ()** – Активирует профилирование запросов, если это разрешено в параметрах.

boolean **get_profiling ()** – Профилирование запросов.

Сессии

Для работы с сессией описан класс *Session* в файле *includes/session.php*. Объект класса создается при инициации сайта и доступен через переменную `$this->diafan->_session`.

Внимание! Методы данного класса являются системными. Их изменение может вызвать нестабильную работу всей системы.

Свойства

var **name** – название сессии.

Пример:

```
echo 'Название сессии: '.$this->diafan->_session->name;
// выведет:
// Название сессии: SESS629188aff6a3c660dc46dedf8724eb61
```

var **id** – идентификатор сессии.

Пример:

```
// используем сессию как идентификатор для поиска
// активированных купонов на скидку для текущего пользователя
$coupon_id = DB::query_result("SELECT coupon_id FROM {shop_discount_person}"
." WHERE session_id='%s'", $this->diafan->_session->id);
```

Методы

void **init ()** – Стартует сессию.

Пример:

```
// запускаем сессию
$this->diafan->_session->init();
```

Функции **open**, **close**, **read**, **write**, **dc**, **destroy** являются пользовательскими функциями хранения сессии. Они подключаются через PHP-функцию [session_set_save_handler](#).

boolean true **open ()** – Открывает сессию.

boolean true **close ()** – Закрывает сессию освобождает ресурсы.

string **read (string \$key)** – Читает сессию.

- string \$key: идентификатор сессии

return true **write (string \$key, string \$value)** – Записывает данные в сессию.

- string \$key: идентификатор сессии
- string \$value: сериализованные данные сессии

void **gc ()** – Чистит мусор - удаляет сессии старше \$lifetime.

void **destroy ([string \$key = ""])** – Удаляет сессию.

- string \$key: идентификатор сессии

Пример:

```
// удаляем сессию, когда пользователь нажал «Выйти»  
$this->diafan->_session->destroy();
```

void **duration ()** – Определяет продолжительность сессии.

Пример:

```
// если при авторизации отмечена опция «Чужой компьютер»,  
// то длительность сессии живет до закрытия окна браузера  
// иначе длительность сессии устанавливается сроком на 2 недели  
$this->diafan->_session->duration();
```

Стандартно куки для сессии хранятся две недели.

void **HTTP_HOST ()** – Возвращает доменное имя.

Сжатие страниц

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

void init () – Инициализирует сжатие. Включается буферизация вывода, определяются и передаются необходимые HTTP заголовки.

void do_gzip () – Выдает сжатые данные, очищает (стирает) буфер вывода и отключает буферизацию вывода.

Пример:

```
Custom::inc('includes/gzip.php');

// стартует процесс записи вывода в буфер
Gzip::init();

// ...
// выполнение программы
// ...

// выдает сжатые данные из буфера
Gzip::do_gzip();
```

Файлы

Для работы с файлами и папками в файле *includes/file.php* определен класс *File*.

Методы

void **check_file (string \$file_path)** – Проверяет существует ли файл.

- string \$file_path: путь до файла относительно корня сайта

Пример:

```
try
{
    File::check_dir("modules/news/news.model.php");
}
catch (Exception $e)
{
    echo $e->getMessage();
}
// если файл не существует, выведет:
// Ошибочный путь.
```

void **copy_file (string \$source, string \$file_path)** – Копирует файл.

- string \$source: полный путь до исходного файла
- string \$file_path: путь до нового файла относительно корня сайта

Пример:

```
// скопируем файл modules/news/news.model.php
// в файл modules/news/news.model.copy.php
File::copy_file(ABSOLUTE_PATH."modules/news/news.model.php",
"modules/news/news.model.copy.php");

// скопируем файл http://www.diafan.ru/img/logo.png
// в файл userfiles/logo_cms_diafan.png
File::copy_file("http://www.diafan.ru/img/logo.png", "userfiles/logo_cms_diafan.png");
```

void **upload_file (string \$tmp_path, string \$file_path)** – Загружает файл и удаляет временный файл.

- string \$tmp_path: полный путь, где храниться временный файл
- string \$file_path: путь до нового файла относительно корня сайта

Пример:

```
// скопируем в файл userfiles/images/new_image.jpg
// только что загруженный через форму файл из переменной image
File::upload_file($_FILES["image"]["tmp_name"], "userfiles/images/new_image.jpg");
```

void **save_file (string \$content, string \$file_path)** – Сохраняет файл.

- string \$content: содержание файла
- string \$file_path: путь до нового файла относительно корня сайта

Пример:

```
// создадим файл userfiles/file.txt с указанным содержимым
File::save_file('Содержимое файла.', "userfiles/file.txt");
```

void **rename_file (string \$name, string \$old_name, string \$path)** – Переименовывает файл.

- string \$name: новое имя
- string \$old_name: старое имя
- string \$path: путь до папки, в которой лежит файл, относительно корня сайта

Пример:

```
// переименуем файл tmp/my_img.jpg
// в файл tmp/my_img_new.jpg
File::rename_file("my_new_img.jpg", "my_img.jpg", "tmp");
```

void **delete_file (string \$file_path)** – Удаляет файл.

- string \$file_path: путь до файла относительно корня сайта

Пример:

```
// удалим файл userfiles/my_file.zip
File::delete_file("userfiles/my_file.zip");
```

mixed File size or false if error **file_size (string \$file_path)** –

Возврат размера файла (даже для файла > 2 Гб) For file size over PHP_INT_MAX (2 147 483 647), PHP filesize function loops from -PHP_INT_MAX to PHP_INT_MAX.

.

- string \$file_path: путь до файла относительно корня сайта

string **hash_file (string \$file_path)** – Возврат хэш файла.

- string \$file_path: путь до файла относительно корня сайта

void **check_dir (string \$dir_path)** – Проверяет существует ли папка.

- string \$dir_path: путь до папки относительно корня сайта

Пример:

```
try
{
    File::check_dir("cache/files");
}
```

```

}
catch (Exception $e)
{
    echo $e->getMessage();
}
// если директория не существует, выведет:
// Ошибочный путь.

```

void **create_dir** (string \$path, [boolean \$access_close = false]) – Создает папку, если она не создана.

- string \$path: путь до папки-родителя относительно корня сайта
- boolean \$access_close: доступ к папке извне будет закрыт

Пример:

```

// создаем папку cache/files
// все файлы в папке будут недоступны извне
File::create_dir("cache/files", true);

```

void **rename_dir** (string \$name, string \$old_name, string \$path) – Переименовывает папку.

- string \$name: новое имя папки
- string \$old_name: старое имя папки
- string \$path: путь до папки-родителя относительно корня сайта

Пример:

```

// переименовываем папку userfiles/my/files
// в userfiles/my/images
File::rename_dir("images", "files", "userfiles/my");

```

void **copy_dir** (string \$source, string \$path) – Копирует папку.

- string \$source: полный путь до исходной папки
- string \$path: путь до папки-назначения относительно корня сайта

Пример:

```

// копируем папу modules/news в custom/my/modules/news
File::copy_dir(ABSOLUTE_PATH."modules/news", "custom/my/modules/news");

```

void **delete_dir** (string \$dir_path) – Удаляет папку.

- string \$dir_path: путь до папки относительно корня сайта

Пример:

```

// удаляем папку cache/files
File::delete_dir("cache/files");

```

boolean **is_writable (string \$path, [boolean \$ftp = false])** – Определяет, доступны ли файл или папка для записи.

- string \$path: путь до файла или папки относительно корня сайта
- boolean \$ftp: учитывать возможность редактирования по FTP

Пример:

```
if(File::is_writable("cache/files"))
{
    echo "папка доступна для записи";
}
// если папка cache/files/ доступна для записи, выведет:
// папка доступна для записи
```

string **compress (string|array \$path, string \$type)** – Сжимает JS и CSS файлы.

- string|array \$path: путь до файла относительно корня сайта
- string \$type: тип: css, js

Пример:

```
// сжимаем файлы css/style.css и css/menu.css и подключаем их
$compress_files = File::compress(['css/style.css', 'css/menu.css'], 'css');

// если включен режим разработки, то файлы не будут сжаты и объединены, тогда
подключаем каждый файл отдельно
if(is_array($compress_files))
{
    foreach($compress_files as $file)
    {
        echo '<link href="'.BASE_PATH.$file.'" rel="stylesheet" type="text/css">';
    }
}
// если выключен режим разработки, то файлы будут сжаты и объединены в один
else
{
    echo '<link href="'.BASE_PATH.$compress_files.'" rel="stylesheet" type="text/css">';
}

// сжимаем файл js/site.js и подключаем его
$path = File::compress('js/site.js', 'js');
echo '<script type="text/javascript" src="'.BASE_PATH.$path.'"></script>';
```

array **rglob (string \$pattern, [string \$flags = 0])** –

Рекурсивный поиск файлов (не происходит раскрытие тильды и подстановка параметров)

.

- string \$pattern: шаблон согласно правилам, используемым в функции glob()

библиотеки libc

- string \$flags: флаги

SMS

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, [string &\$error_output = "], [string &\$trace_output = "]) – Отправляет SMS.

- string \$text: текст SMS
- string \$to: номер получателя
- string \$error_output: вывод ошибки
- string \$trace_output: вывод трассировки

Пример:

```
// отправляем SMS
Custom::inc('includes/sms.php');
Sms::send('На сайте появился новый заказ №134.', '79999999999');
```

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;
- [JsHttpRequest](#) – заимствовано преобразование данных в формат JSON;
- [KCAPTCHA](#) и [RECAPTCHA](#) – 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;
- [Minify](#) – сжатие CSS и JS файлов;
- [Mobile Detect](#) – определение мобильных устройств;
- [jQuery](#) – JavaScript библиотека;
- [jQuery UI](#) – JavaScript библиотека для создания насыщенного пользовательского интерфейса в веб-приложениях, часть проекта jQuery;
- [jQuery Form Plugin](#) – jQuery плагин для работы с формами;
- [prettyPhoto](#) – jQuery плагин для анимации открытия изображений;
- [jQuery Tooltip plugin](#) – jQuery плагин всплывающих подсказок;
- [jQuery File Upload Plugin](#) – jQuery плагин для мультizaгрузки файлов;
- [Superfish](#) – jQuery плагин для создания выпадающего меню;
- [hoverIntent by Brian Cherne](#) – jQuery плагин для создания выпадающего меню;
- [jCarouselLite](#) – jQuery плагин для прокрутки слайдов;
- [imgAreaSelect](#) – jQuery плагин для выделения области на изображении;
- [timePicker](#) – jQuery плагин календаря для поля даты;
- [jQuery.ScrollTo](#) – программная перемотка скrolла;
- [jQuery Masked Input](#) – маски ввода для полей формы;
- [extsrcjs](#) – асинхронная загрузка JS-файлов.

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](#)
- [Изменение административной части](#)
- [Как в поиске по товарам заменить чекбоксы на список](#)
- [Как вывести дополнительную характеристику товара в любом месте](#)
- [Как добавить виджет комментариев из социальной сети](#)
- [Частичная кастомизация](#)

Как создать свой модуль

- [Создаем таблицу в БД](#)
- [Панель администрирования](#)
- [Вкладки для части модуля и конфигурация](#)
- [Пользовательская часть](#)

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](#).

Модуль – набор файлов в папке *modules/имя_модуля*. [Подробнее об устройстве модуля](#).

В нашем примере модуль будет называться *example*. Все файлы модуля будут храниться в папке *modules/example*.

Для того чтобы подключить наш сырой модуль к DIAFAN.CMS, необходимо сделать записи в таблицах [{modules}](#) и [{admin}](#). Для этого в файле *example.install.php* прописываем:

Пример:

```
// записи в таблице {modules}
public $modules = array(
    array(
        "name" => "example",
        "admin" => true,
        "site" => true,
        "site_page" => true,
    ),
);

// меню административной части
public $admin = array(
    array(
        "name" => "Название модуля",
        "rewrite" => "example",
        "group_id" => "1",
        "sort" => 5,
        "act" => true,
        "children" => array(
            array(
                "name" => "Настройки",
                "rewrite" => "example/config",
            ),
        ),
    ),
);
```

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 *ВАШ_ПРЕФИКС_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
<input type="checkbox"/>	id	int(11)		UNSIGNED	Нет	Нет	AUTO_INCREMENT
<input type="checkbox"/>	user_id	int(11)		UNSIGNED	Нет	0	
<input type="checkbox"/>	text	text	utf8_general_ci		Нет	Нет	
<input type="checkbox"/>	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:

```
public $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',
            ),
        ),
    );

    // описание полей в списке
    public $variables_list = array (
        'checkbox' => "",
        'name' => array(
            'name' => 'Текст объявления',
        ),
    );

    // ссылка на добавление нового объявления
    public function show_add()
    {
        $this->diafan->addnew_init('Добавить объявление');
    }

    // функция, которая определяет что выводит модуль при открытии
    public function show()
    {
        // список объявлений
        $this->diafan->list_row();
    }
}

```

Свойство класса `$table` определяет в какой таблице будут храниться наши объявления, массив `$variables` описывает каждое поле таблицы, массив `$variables_list` описывает какие поля и как будут выводиться в списке, 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 дополнительный атрибут [select_db](#), в котором указываем, в какой таблице базы данных и какие данные брать.

Пример:

```
// описание полей таблицы
public $variables = array (
    'main' => array (
        'created' => array(
            'type' => 'datetime',
            'name' => 'Дата создания',
        ),
        'user_id' => array(
            'type' => 'select',
            'name' => 'Автор',
            'select_db' => array(
                'table' => 'users',
                'name' => 'fio',
                'where' => "trash='1'",
            ),
        ),
        'text' => array(
            'type' => 'editor',
            'name' => 'Текст объявления',
        ),
    ),
);
```

Теперь опишем как будет выглядеть список объявлений. Для этого нужно их добавить в массив [\\$variables_list](#).

Пример:

```
// описание полей в списке
public $variables_list = array (
    'checkbox' => "",
    'created' => array(
        'name' => 'Дата и время',
        'type' => 'datetime',
        'sql' => true,
    ),
    'name' => array(
        'name' => 'Текст объявления',
    ),
    'user_id' => array(
        'name' => 'Пользователь',
        'type' => 'string',
    ),
);
```

```

        'sql' => true,
    ),
    'actions' => array(
        'del' => true,
    ),
);

```

Мы добавили два поля из базы данных (атрибут *sql* означает, что будет брать данные из базы). И указали формат, в котором данные выводятся. И добавили действие «Удалить». И последнее. Мы должны указать, что основное поле у нас называется *text* (по умолчанию *name*):

Пример:

```

'name' => array(
    'name' => 'Текст объявления',
    'variable' => 'text',
),

```

Все, этого вполне достаточно чтобы выводить список объявлений, добавлять, редактировать и сохранять их.

Ура, теперь у нас в списке выводится объявление, время добавления и *id* пользователя.

Дата и время	Текст объявления	Польз
<input type="checkbox"/> 12.12.2015 23:04	Воротник-стойка защитит от попадания холодного воздуха. Низ куртки регулируется эластичным шнуром. Интересный дизайн подчеркнёт Вашу индивидуальность.	
<input type="checkbox"/> 12.12.2015 15:30	На 4 человек, желательно раздельную 2+2.	
<input type="checkbox"/> 06.11.2015 22:20	Компактный среднеразмерный рюкзак для любителей охоты и рыбалки. В двух больших боковых карманах и одном с фронтальной части, без труда разместятся...	

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: *list_variable_переменная(\$row)*, где *\$row* – массив значений для текущего элемента:

Пример:

```

public function list_variable_user_id($row)
{
    return '<div>'.DB::query_result("SELECT fio FROM {users} WHERE id=%d", $row['user_id']).'</div>';
}

```

То есть, если DIAFAN.CMS обнаруживает эти функции с названием по маске *list_variable_**,

она понимает, что поле `user_id` нужно выводить так, как описано в этой функции.

Дата и время	Текст объявления	Пользователь
<input type="checkbox"/> 12.12.2015 23:04	Воротник-стойка защитит от попадания холодного воздуха. Низ куртки регулируется эластичным шнуром. Интересный дизайн подчеркнёт Вашу индивидуальность.	Михаил Волков
<input type="checkbox"/> 12.12.2015 15:30	На 4 человек, желательно раздельную 2+2.	Иванов Иван Ивано
<input type="checkbox"/> 06.11.2015 22:20	Компактный среднеразмерный рюкзак для любителей охоты и рыбалки. В двух больших боковых карманах и одном с фронтальной части, без труда разместятся...	Михаил Волков

Внимание! Пример показывает как использовать функцию `list_variable_()`. *Оптимальнее для этой задачи использовать тип `*select` с описанным уже в `$variables` атрибутом `select_db`.*

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 `$variables_list`.

Пример:

```
// описание полей в списке
public $variables_list = array (
    'checkbox' => "",
    'sort' => array(
        'name' => 'Сортировка',
        'type' => 'numtext',
        'desc' => true,
        'sql' => true,
        'fast_edit' => true,
    ),
    'created' => array(
        'name' => 'Дата и время',
```



```

        'type' => 'datetime',
        'sql' => true,
    ),
    'name' => array(
        'name' => 'Текст объявления',
    ),
    'user_id' => array(
        'name' => 'Пользователь',
        'type' => 'string',
        'sql' => true,
    ),
    'actions' => array(
        'act' => true,
        'trash' => true,
    ),
);

```

Теперь у нас в списке объявлений появятся иконки удаления, активности, и их можно будет сортировать.

Внимание!

Также не стоит забывать про переменную [\\$variables_filter](#), которая позволяет нам добавлять собственные параметры в фильтре.

Давайте с ее помощью для примера организуем фильтр объявлений по имени пользователя.

Пример:

```

public $variables_filter = array (
    'user_id' => array(
        'type' => 'select',
        'name' => 'Искать по пользователю',
    ),
);

```

Значения для списка берем из базы данных. Таблицу базы данных берем из атрибута `select_db` в массиве `$variables`.

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*.

Пример:

```
public function edit_variable_user_id()
{
    echo '<div class="unit">
        <div class="infofield">Нажми</div>
        <div class="user_id" rel="'. $this->diafan->value.' "><b>ЗДЕСЬ</b></div>
    </div>';
}
```

Напишем простой jQuery обработчик, отправляющий Ajax-запрос. Для этого создадим файл *modules/example/admin/js/example.admin.edit.js*. Этот JS-файл подключиться для формы редактирования автоматически. Созданный файл будет следующего содержания:

Пример:

```
$(".user_id").click(function(){
    var user_id = $(this).attr('rel');
    diafan_ajax.init({
        data:{
            action: "user",
            module: "example",
```

```

        user_id: user_id
    },
    success: function(response) {
        alert(response.name);
    }
});
});

```

Обработчик по нажатию на элемент с классом *user_id* отправляет AJAX-запрос текущей странице с параметрами *module*, *action* и *user_id*. Аjax-запрос отправляем через обработчик *diafan_ajax*, чтобы были добавлен идентификационный хэш и его проверка.

Теперь нам нужно написать *example.admin.action.php*

Пример:

```

class Example_admin_action extends Action_admin
{
    public function init()
    {
        if (! empty($_POST['user_id']))
        {
            $this->result["name"] = DB::query_result("SELECT fio FROM {users} WHERE id=%d",
            $_POST['user_id']);
        }
        else
        {
            $this->result["name"] = 'ошибка';
        }
    }
}

```

Файл *example.admin.action.php* подключается лишь тогда, когда одновременно отправляются POST переменные *action* и *module*. Значение переменной *module* должно соответствовать названию нашего модуля. При соблюдении этих условий подключиться функция *init()*, в которой должно быть описано обработка запроса. Данные, сохраненные в переменной *\$this->result* автоматически будут отправлены обратно в JSON-формате.

Теперь доработаем функцию *save_variable_user_id*, в которой происходит 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:

```

public function save_variable_user_id()
{
    $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.

Текущую выборку из базы данных можно получить через функцию `$this->diafan->values()`. Например, `$this->diafan->values('site_id')` вернет значение поля *site_id*.

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*, например, при редактировании новостей в файле *modules/news/admin/news.admin.php* в `$variables` прописываем.

Пример:

```
public $variables = array(  
    'main' => array(  
        'example' => 'module',  
        ...  
    ),  
    ...  
);
```

При редактировании новости 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/admin/* и добавляем подраздел нашему модулю с псевдоссылкой *example/bolt*. Количество расширений модуля зависят лишь от вашей фантазии.

Есть еще особый тип расширения – **конфигурация**.

Регистрируется он так же, как и обычное расширение: в *http://site.ru/admin/admin/* 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 контроллера `example.php` вызывается функция `init()`.

Как можно видеть из примера, при инициализации модуля подключается модель. Результат исполнения записывается в переменную `$this->result`. После подключается шаблон, указанный в `$this->result["view"]`. В примере `$this->result["view"] = 'show'`, значит подключается файл `modules/example/views/example.view.show.php`.

Если в модуле будут использоваться переменные, передаваемые в URL, то их нужно указать в массиве `rewrite_variable_names`.

Пример:

```
public $rewrite_variable_names = array('page', 'show');
```

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->nen = DB::query_result("SELECT COUNT(id) FROM {example}
        WHERE act='1' AND trash='0'");
        $this->result["paginator"] = $this->diafan->_paginator->get();
    }
}
```

```

////navigation///

$rows = DB::query_range_fetch_all("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);

foreach($rows as $row)
{
    $row['created'] = $this->format_date($row['created'], 'example');
    $row['link'] = $this->diafan->_route->link($this->diafan->_site->id, $row["id"],
'example');

    $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"]);
$this->result['view'] = 'show';
}

```

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->_site->id,
);
// 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](#).

Функции `format_date()` форматирует дату в соответствии с настройкой модуля, она наследуется из класса `Model`, подробнее о ней можно почитать [здесь](#).

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',
            'select' => array(
                0 => '1.05.2014',
                1 => '1 мая 2014 г.',
                2 => '1 мая',
                3 => '1 мая 2014, понедельник',
                4 => 'не отображать',
            ),
        ),
    ),
);

public $config = array('config');
```

Получился функциональный модуль аж с двумя изменяемыми настройками и кэшированием.

Вы можете создать также отдельный шаблонный тег для модуля. Об этом есть отдельный урок [«Как добавить шаблонный тег»](#).

[Исходный код модуля example.](#)

Как добавить шаблонный тег

Шаблонные теги позволяют подключать исполняемый код в определенное место сайта.

DIAFAN.CMS не делает никаких ограничений на количество аргументов и название функции. Исполняемый код может быть общего характера или относиться к модулю.

Если нужно добавить **код общего характера**, то создаем общий шаблонный тег. Для этого в папку *themes/functions* нужно добавить файл с названием тега. Стоит заметить, что имеется ввиду папка

/custom/ваша_тема/themes/functions

поскольку все изменения следует вносить в файлы пользовательских тем ([о темах](#)).

После этого шаблонный тег можно использовать.

Тегу можно передавать атрибуты тега. Атрибуты будут доступны через массив *\$attributes*.

Пример:

Добавляем файл *hello.php*.

```
<?php
if (! defined('DIAFAN'))
{
    $path = __FILE__; $i = 0;
    while(! file_exists($path.'/includes/404.php'))
    {
        if($i == 10) exit; $i++;
        $path = dirname($path);
    }
    include $path.'/includes/404.php';
}
echo "Hello, ".$attributes["username"].!"";
```

Затем в любом основном шаблоне сайта, например в *themes/site.php*, файл можно исполнить в любом месте шаблонным тегом

```
<insert name="hello" username="Пользователь">
```

Пример:

Добавляем модулю «Новости» шаблонную функцию *show_date*

```
// в файле modules/news/news.php
public function show_date($attributes)
{
    $attributes = $this->get_attributes($attributes, 'day', 'template');
```



```
$result = $this->model->date($attributes["day"]);  
echo $this->diafan->_tpl->get('show_date', 'news', $result, $attributes["template"]);  
}
```

// в файле *modules/news/news.model.php*

```
public function date($day)
```

```
{  
    if($day)  
    {  
        $result["date"] = $day.date(".m.Y");  
    }  
    else  
    {  
        $result["date"] = date("d.m.Y");  
    }  
    return $result;  
}
```

// файл *modules/news/views/news.view.show_date.php*

```
echo '<div class="news_date">'.$result["date"].'</div>';
```

Затем в любом основном шаблоне сайта, например в *themes/site.php*, функцию можно вызвать в любом месте шаблоном 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:

```
$rows = DB::query_range_fetch_all("SELECT e.id, e.author, e.created, ...");
```

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:

```
$rows = DB::query_range_fetch_all("SELECT e.id, e.author, e.[name], ...");
```

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::query_fetch_array("SELECT id, author, [name], [anons], ...");
```

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["rows"] = DB::query_range_fetch_all("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.

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 папке *themes/functions*. Добавим наш файл для шаблонного тега. Назовем его *show_last_comments.php*:

Пример:

```
<?php  
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:

```
$rows = DB::query_fetch_all("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 файл примет такой вид:

Пример:

```
<?php  
$rows = DB::query_fetch_all("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");  
$text = "";  
foreach ($rows as $row)  
{  
    $text .= $row['text'].'<br>';  
}  
echo $text;
```

Но в таком виде он ничего не выводит. Давайте выведем текст комментариев:

Пример:

```
$text = "";
foreach ($rows as $row)
{
    $text .= $row['text'].'<br>';
}
echo $text;
```

Исходный код стал выглядеть так:

Пример:

```
<?php
$rows = DB::query_fetch_all("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");
$text = "";
foreach ($rows as $row)
{
    $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 код:

Пример:

```

<?php
$rows = DB::query_fetch_all("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");
$text = "";
foreach ($rows as $row)
{
    if ($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;

```

В начале файла добавим следующий код для исключения прямого доступа к файлу.

Пример:

```

<?php
if (! defined('DIAFAN'))
{
    $path = __FILE__; $i = 0;
    while(! file_exists($path.'/includes/404.php'))
    {
        if($i == 10) exit; $i++;
        $path = dirname($path);
    }
    include $path.'/includes/404.php';
}

```

Получаем итоговый код:

Пример:

```

<?php
if (! defined('DIAFAN'))
{
    $path = __FILE__; $i = 0;
    while(! file_exists($path.'/includes/404.php'))
    {
        if($i == 10) exit; $i++;
        $path = dirname($path);
    }

```

```

    }
    include $path.'/includes/404.php';
}

$rows = DB::query_fetch_all("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");
$text = "";
foreach ($rows as $row)
{
    if ($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 папке *themes/functions* добавим еще один файл. Например, *show_my_code.php*.

Пример:

```
<?php
if (! defined('DIAFAN'))
{
    $path = __FILE__; $i = 0;
    while(! file_exists($path.'/includes/404.php'))
    {
        if($i == 10) exit; $i++;
        $path = dirname($path);
    }
    include $path.'/includes/404.php';
}
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 файл *show_my_code.php* будет подключаться 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 Ајах-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 displays the 'Бумажный зоопарк' (Paper Zoo) catalog management interface. A new characteristic is being added with the following details:

- Название (Name):** Материк (Continent)
- Тип (Type):** выпадающий список (Dropdown list)
- Значения, псевдоссылка (Values, pseudolink):** A list of six continents: Евразия, Северная Америка, Южная Америка, Африка, Австралия, and Антарктида. Each value is in a separate input field.
- Отдельная страница для значений (Separate page for values):** ☐
- Категория (Category):** A dropdown menu with options: Птицы, Звери, Крупные, and Мелкие.
- Сортировка: установить перед (Sorting: set before):** Материк
- Использовать в форме поиска (Use in search form):** ☒
- Показывать в списке (Show in list):** ☐
- Показывать в блоке товаров (Show in product block):** ☐
- Описание характеристики (Characteristic description):** A large empty text area at the bottom.

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 `diafan_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->id == 12)  
{  
    $mainland = array();  
    $rows = DB::query_fetch_all("SELECT id, [name] FROM {shop_param_select} WHERE  
param_id=%d", 11);  
    foreach($rows as $row)  
    {  
        $mainland[$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->id == 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 *modules/shop/js/shop.show_search.js* и в самый конец 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.action.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.

Сначала в файле *modules/shop/shop.php* в метод *action()* добавим после

Пример:

```
switch ($_POST['action'])
{
```

добавим

Пример:

```
case 'change_country':
return $this->action->change_country();
```

Теперь в файле *modules/shop/shop.action.php* добавим новый метод:

Пример:

```

public function change_country()
{
    $text = '<option value="0">-</option>';
    $rows = DB::query_fetch_all("SELECT id, [name] FROM {shop_param_select} WHERE
parent_param_id=%d", $_POST['mainland']);
    foreach($rows as $row)
    {
        $text .= '<option value="'. $row['id']. ">' . $row['name']. '</option>';
    }
    echo $text;
}

```

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 `ВАШ_ПРЕФИКС_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 '  
<div class="unit" id="photo">  
    <div class="infofield">'. $this->diafan->variable_name(). '</div>';  
  
    $photos = DB::query_fetch_all("SELECT id, [name] FROM {photo} WHERE [act]='1' AND  
trash='0'");  
  
    echo '<select name="photo">';  
    echo '<option value="0">-</option>';  
    foreach($photos as $photo)  
    {  
        echo '<option value="'. $photo['id']. '"'. ($this->diafan->value == $photo['id'] ? 'selected' :  
        ""). '>'. $photo['name']. '</option>';  
    }  
}
```

```

    }
    echo '</select>
</div>';
}

```

Рассмотрим некоторые ключевые моменты при создании подобных методов.

Во-первых, это его именование. Оно обязательно должно быть вида `edit_variable_имя_поля`.

Во-вторых, описание текущей переменной из массива `$variables` можно получить через функцию `$this->diafan->variable_name()`. Через эту же функцию можно получить описание любой переменной, передав название переменной в качестве аргумента. Например, `$this->diafan->variable_name('timeedit')`.

В-третьих, это цикл, в котором выводятся все элементы в виде выпадающего списка. В нем мы сравниваем значение идентификатора фотографии и текущее значение поля `photo` – `$this->diafan->value`, чтобы добавить атрибут "selected" пункту выпадающего списка, если значение поля и идентификатора совпадают. Все текущие значения полей можно получить через функцию `$this->diafan->values()`. Мы могли бы использовать `$this->diafan->values('photo')` вместо `$this->diafan->value`.

Результат можно увидеть на скриншоте:

Второй метод, который нам необходим, выглядит так:

Пример:

```

public function save_variable_photo()
{
    $this->diafan->set_query("photo=%d");
    $this->diafan->set_value($_POST['photo']);
}

```

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 предельно проста – сохраняем выбранное пользователем значение выпадающего списка фотографий в поле `photo` для текущей статьи.

Функция `set_query()` указывает в какое поле таблицы `{clauses}` в базе данных сохранять значение и какая [маска для фильтрации](#).

Функция `set_value()` указывает какое значение сохранять.

Далее выведем фотографию на подробной странице статьи.

Открываем файл `modules/clauses/clauses.model.php` и вносим изменения в метод `id_query`:

Пример:

```
$row = DB::query_fetch_array("SELECT id, photo, [name], [anons], ...");
```

а также добавляем в метод `id` следующий код:

Пример:

```
$photo = $this->diafan->_images->get('medium', $this->result["photo"], 'photo', 'element', 0,
"");

if($photo)
{
    $this->result["photo"] = $photo[0];
}
```

Методом `$this->diafan->_images->get` получаем изображение фотографии. 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:

Палатки «Каван 2, 3, 4». Инструкция по эксплуатации



27 мая

Пожалуйста, ознакомьтесь с данной инструкцией, прежде чем устанавливать палатку и сохраните ее для справок в дальнейшем. Палатка «Каван» идеально подойдет для комфортного отдыха или для использования в качестве базовой. Палатка устанавливается на разборном дуговом каркасе, имеет плотное непромокаемое дно, большой тамбур и два входа на молнии.



[Вернуться к списку](#)

[Палатки «Гори 2, 3, 4» Инструкция по эксплуатации](#) →

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](#)).

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](#).

Файл *includes/mail.php* подключаем через функцию `inc()` в классе `Custom`, чтобы можно было использовать этот метод из кастомизированной темы. Подробнее в [разделе «Кастомизация»](#).

Пример:

```
Custom::inc("includes/mail.php");
```

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 (если предустановленные данные о типах пользователей не были изменены). Также нужно добавить условие «активен на сайте» и «не удален в корзину». Зная это, мы можем выбрать нужные нам почтовые ящики.

Пример:

```
$mails = DB::query_fetch_value("SELECT DISTINCT(mail) FROM {users} WHERE role_id=3 AND act='1' AND trash='0'", "mail");
```

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 сохраняемую новость.

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 и ID текущей новости.

ID страницы получаем методом `get_site_id()`.

ID новости хранится в `$this->diafan->id`.

Модуль, в нашем случае, имеет название "news".

В итоге получим такой код:

Пример:

```
$site_id = $this->diafan->get_site_id();  
$link = $this->diafan->_route->link($site_id, $this->diafan->id, "news");
```

Теперь остается циклом перебрать все почтовые ящики администраторов и отправить им уведомления. Результат будет выглядеть так:

Пример:

```
public function save_variable_sendmail()  
{  
    Custom::inc("includes/mail.php");  
  
    //тема письма  
    $subject = 'Изменения на сайте '.TIT1;  
  
    //ссылка на текущую новость
```

```
$site_id = $this->diafan->get_site_id();
$link = BASE_PATH.$this->diafan->_route->link($site_id, $this->diafan->id, "news");

//текст сообщения
$message = '<a href="'. $link. "'>'.(! empty($_POST['name']) ? $_POST['name'] : $this->diafan->id). '</a>';

//выбираем значения почтовых ящиков администраторов
$mails = DB::query_fetch_value("SELECT DISTINCT(mail) FROM {users} WHERE role_id=3 AND act='1' AND trash='0'", "mail");

//делаем рассылку
foreach($mails as $mail)
{
    send_mail($mail, $subject, $message);
}
}
```

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 доставки 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 `$variables_list`.

Добавим в него новый элемент `delivery_id`. В результате массив примет такой вид:

Пример:

```
public $variables_list = array (
    'checkbox' => "",
    'created' => array(
        'name' => 'Дата и время',
        'type' => 'datetime',
        'sql' => true,
        'no_important' => true,
    ),
    'name' => array(
        'name' => 'Заказ',
        'variable' => 'id',
        'text' => '№ %d'
    ),
    'status_id' => array(
        'name' => 'Статус',
        'sql' => true,
    ),
    'delivery_id' => array(
        'name' => 'Доставка',
        'sql' => true,
    ),
    'summ' => array(
        'name' => 'Сумма',
        'sql' => true,
    ),
    'user_id' => array(
        'name' => 'Покупатель',
        'sql' => true,
    ),
    'actions' => array(
        'trash' => true,
    ),
)
```



```
);
```

Добавим метод, который формирует выводимые данные для нового поля.

Пример:

```
public function list_variable_delivery_id($row, $var)
{
    $delivery = DB::query_result("SELECT [name] FROM {shop_delivery} WHERE id=%d",
    $row['delivery_id']);
    return '<div>'.$delivery.'</div>';
}
```

Рассмотрим метод подробнее.

Обязательно следует обратить внимание на именование метода. Он всегда начинается с *list_variable_*, а далее идет значение ключа элемента массива *\$variables_list*. В нашем случае это *delivery_id*.

В метод передается массив *\$row*, который содержит ряд параметров текущего заказа, в том числе нужный нам *\$row['delivery_id']*.

\$row['delivery_id'] — это ID способа доставки таблицы *{shop_delivery}*. Он позволяет нам выбрать имя способа доставки для конкретного заказа. Сохраняем имя в переменной *\$delivery*.

Далее мы просто выводим его в общем списке.

Наш метод будет выполняться для каждого заказа из списка. При этом каждый раз будет выполняться SQL-запрос. Оптимизируем код так, чтобы в начале выполнялся один SQL-запрос с поиском всех используемых способов доставки, а затем эти данные просто будут использоваться.

Сначала запросим все используемые способов доставки и запишем эту информацию в переменную *\$this->cache["prepare"]["delivery"]*.

Пример:

```
// если данные еще не запрашивались
if(! isset($this->cache["prepare"]["delivery"]))
{
    // собираем ID способов доставки из заказов, выведенных в списке
    $delivery_ids = array();
    foreach($this->diafan->rows as $r)
    {
        if($r["delivery_id"] && ! in_array($r["delivery_id"], $delivery_ids))
        {
            $delivery_ids[] = $r["delivery_id"];
        }
    }
    if($delivery_ids)
```

```

{
    // запрашиваем информацию о найденных способах доставки
    // результат записываем в виде массива с ключами,
    // где ключами будет поле id, а значениями поля name
    $this->cache["prepare"]["delivery"] = DB::query_fetch_key_value(
        "SELECT id, [name] FROM {shop_delivery} WHERE id IN (%s)",
        implode(";", $delivery_ids),
        "id", "name"
    );
}
}

```

Затем просто используем полученные данные.

Пример:

```

if($row["delivery_id"] && ! empty($this->cache["prepare"]["delivery"][$row["delivery_id"]]))
{
    echo $this->cache["prepare"]["delivery"][$row["delivery_id"]];
}

```

Итоговый код:

Пример:

```

public function list_variable_delivery_id($row)
{
    if(! isset($this->cache["prepare"]["delivery"]))
    {
        $delivery_ids = array();
        foreach($this->diafan->rows as $r)
        {
            if($r["delivery_id"] && ! in_array($r["delivery_id"], $delivery_ids))
            {
                $delivery_ids[] = $r["delivery_id"];
            }
        }
        if($delivery_ids)
        {
            $this->cache["prepare"]["delivery"] = DB::query_fetch_key_value(
                "SELECT id, [name] FROM {shop_delivery} WHERE id IN (%s)",
                implode(";", $delivery_ids),
                "id", "name"
            );
        }
    }
    $text = '<div>';
    if($row["delivery_id"] && ! empty($this->cache["prepare"]["delivery"][$row["delivery_id"]]))
    {
        $text .= $this->cache["prepare"]["delivery"][$row["delivery_id"]];
    }
}

```

```
}  
$text .= '</div>';  
return $text;  
}
```

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="input-title">'.$row["name"].':</span>;
    foreach ($row["select_array"] as $key => $value)
    {
        echo '<input type="checkbox" id="shop_search_p'.$row["id"].'_'.$key.'" name="p'.
        $row["id"].'[]" value="'.$key.'"'.(in_array($key, $row["value"]) ? " checked" : "").'>
        <label for="shop_search_p'.$row["id"].'_'.$key.'">'.$value.'</label>
        <br>;
    }
```

Заменяем на следующий код:

Пример:

```
case 'select':
case 'multiple':
    echo '
    <span class="input-title">'.$row["name"].':</span>
    <select name="p'.$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 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 the Facebook Comments widget. It contains 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'));
```

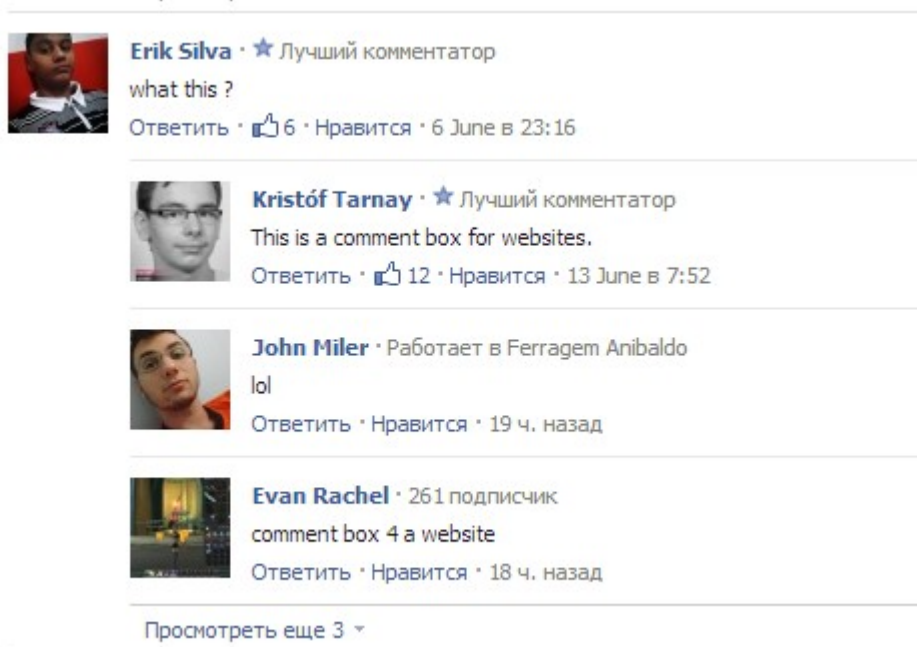
Вторая часть размещается в том месте, где непосредственно будет вывод блока с комментариями.

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.

Сайт/приложение: Подключить новый сайт ▼

Название сайта:

Адрес сайта:

Основной домен сайта:

Сохранить

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>
```

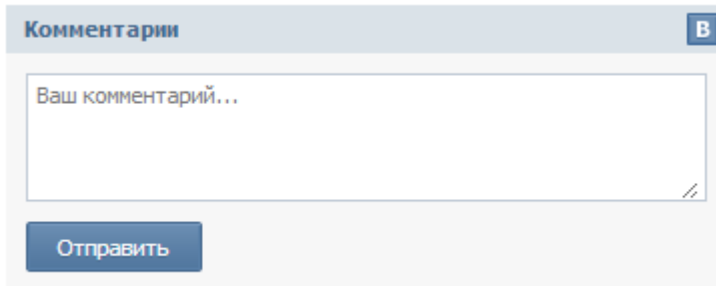
Вторая часть ставится в то место, где будет выводиться контент виджета.

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). The form is styled with a light blue header and a white body.

Оформление.

Оба виджета можно оформить в виде шаблонных тегов. Чтобы сделать это, создадим два новых файла.

themes/functions/facebook.php

Пример:

```
<?php
if (! defined('DIAFAN'))
{
    $path = __FILE__; $i = 0;
    while(! file_exists($path.'/includes/404.php'))
    {
        if($i == 10) exit; $i++;
        $path = dirname($path);
    }
    include $path.'/includes/404.php';
}
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>";
```

themes/functions/vk.php

Пример:

```
<?php

if (! defined('DIAFAN'))
{
    $path = __FILE__; $i = 0;
    while(! file_exists($path.'/includes/404.php'))
    {
        if($i == 10) exit; $i++;
        $path = dirname($path);
    }
    include $path.'/includes/404.php';
}

echo '<script type="text/javascript" src="//vk.com/js/api/openapi.js?63"></script>
<script type="text/javascript">
    VK.init({apild: тут будет твой ap, 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="facebook">
<insert name="vk">
```

Частичная кастомизация файлов

Общие принципы работы частичной кастомизации 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 к 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*.

Приставка *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('a', '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)  
    {  
        $text = str_replace($this->ru, $this->tr, $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.

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

- [custom](#) – выводит путь до файла с учетом кастомизации;
- [language](#) – выводит номер текущей языковой версии сайта;
- [module](#) – выводит подключенный к странице модуль;
- [path](#) – выводит адрес сайта;
- [path_url](#) – выводит адрес сайта, с учетом языковой версии сайта;
- [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_breadcrumb](#) – выводит навигацию по сайту «Хлебные крошки»;
- [show_css](#) – подключает CSS-файлы. При включенном режиме разработки файлы будут объединены и сжаты, что приведет к более быстрой загрузке файлов. Если существуют какие-то проблемы при включенном сжатии, подключите CSS-файлы стандартным HTML-тегом `<link rel="stylesheet" type="text/css" ...>`;
- [show_description](#) – выводит мета-тег description страницы;
- [show_h1](#) – выводит заголовок страницы, если не запрещен его вывод в настройке странице «Не показывать заголовок»;
- [show_head](#) – формирует часть HTML-шапки сайта. Включает в себя в том числе теги: show_title, show_description, show_keywords;
- [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 ``, но ликвидирует ссылки, которые никуда не ведут;
- [show_include](#) – подключает файл-блок шаблона;
- [show_js](#) – подключает JS-файлы. Тег нужно добавить перед `</body>`;
- [show_keywords](#) – выводит ключевые слова страницы. Используется для мета-тега keywords;
- [show_module](#) – выводит контент модуля;
- [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;
- [title](#) – название сайта, заданное в параметрах сайта, на текущем языке.

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.

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](#) – выводит информацию о товарах в списке пожеланий;
- [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_brand](#) – выводит несколько производителей;
- [show_category](#) – выводит несколько категорий;
- [show_last_order](#) – выводит информацию о последнем совершенном заказе;
- [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 артикулу», «Искать товары по акции», «Искать по новинкам», «Искать по хитам»). 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.

Отзывы

- [show](#) – выводит отзывы и форму добавления отзывов для текущей страницы;
- [show_block](#) – выводит последние отзывы.

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 блока на сайте, номер которой передан в виде атрибута id;
- [show_comments](#) – выводит комментарии, прикрепленные к странице сайта, если в конфигурации модуля «Страницы сайты» подключены комментарии;
- [show_dynamic](#) – выводит содержимое динамического блока, номер которой передан в виде атрибута id;
- [show_images](#) – выводит изображения, прикрепленные к странице сайта, если в конфигурации модуля «Страницы сайты» 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-якоря), 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.

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:

- **defer** – маркер отложенной загрузки шаблонного тега: **event** – загрузка контента только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** – загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка контента только по желанию пользователя;
- **defer_title** – текстовая строка, выводимая на месте появления загружаемого контента с помощью отложенной загрузки шаблонного тега;
- **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 шаблоне тега можно получить значение любого атрибута через переменную `$result["attributes"]["название атрибута"]`.

B

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. Можно указать отрицательное значение, тогда будут исключены объявления из указанного раздела. По умолчанию выбираются все страницы;
- **cat_id** – категории объявлений, если в настройках модуля отмечено «Использовать категории». Идентификаторы категорий перечисляются через запятую. Можно указать отрицательное значение, тогда будут исключены объявления из указанной категории. Можно указать значение **current**, тогда будут показаны объявления из текущей (открытой) категории или из всех категорий, если ни одна категория не открыта. По умолчанию категория не учитывается, выводятся все объявления;
- **sort** – сортировка объявлений: **date** – по дате (по умолчанию), **rand** – в случайном порядке;
- **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 `param="3=5&3=6&10>50&16=1"` расшифровывается как объявления о квартире и доме (5 и 6 номер), площадью более 50 кв.м, в новостройке.

```
<insert name="show_block" module="ab" param="3=5&3=6&10>50&16=1">
```

Номер (или идентификатор) характеристики можно посмотреть, если подвести курсор к названию характеристики в списке характеристик в административной части. Появится всплывающая подсказка «Редактировать (номер

характеристики)».

Номер (или идентификатор) значения характеристики можно посмотреть, если при редактировании характеристики подвести курсора на нужное значение. Появится всплывающая подсказка «ID: номер».

- **only_module** – выводить блок только на странице, к которой прикреплен модуль «Объявления»: **true** – выводить блок только на странице модуля, по умолчанию блок будет выводиться на всех страницах;
- **tag** – тег, прикрепленный к объявлениям;
- **defer** – маркер отложенной загрузки шаблонного тега: **event** – загрузка контента только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** – загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка контента только по желанию пользователя;
- **defer_title** – текстовая строка, выводимая на месте появления загружаемого контента с помощью отложенной загрузки шаблонного тега;
- **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

В шаблоне тега можно получить значение любого атрибута через переменную `$result["attributes"]["название атрибута"]`.

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: по умолчанию как в панели администрирования, **date** – по дате, **rand** – в случайном порядке;
- **cat_id** – категория баннеров, если в настройках модуля отмечено «Использовать категории»;
- **defer** – маркер отложенной загрузки шаблонного тега: **event** – загрузка контента только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** – загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка контента только по желанию пользователя;
- **defer_title** – текстовая строка, выводимая на месте появления загружаемого контента с помощью отложенной загрузки шаблонного тега;
- **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" sort="rand">
```

выведет 1 случайный баннер

show_block module="cart"

Выводит информацию о заказанных товарах, т. н. корзину.

Атрибуты:

- **defer** – маркер отложенной загрузки шаблонного тега: **event** – загрузка контента только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** – загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка контента только по желанию пользователя;
- **defer_title** – текстовая строка, выводимая на месте появления загружаемого контента с помощью отложенной загрузки шаблонного тега;
- **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

В шаблоне тега можно получить значение любого атрибута через переменную `$result["attributes"]["название атрибута"]`.

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. Можно указать отрицательное значение, тогда будут исключены статьи из указанного раздела. По умолчанию выбираются все страницы;
- **cat_id** – категории статей, если в настройках модуля отмечено «Использовать категории». Идентификаторы категорий перечисляются через запятую. Можно указать отрицательное значение, тогда будут исключены статьи из указанной категории. Можно указать значение **current**, тогда будут показаны статьи из текущей (открытой) категории или из всех категорий, если ни одна категория не открыта. По умолчанию категория не учитывается, выводятся все статьи;
- **sort** – сортировка статей: по умолчанию как на странице модуля, **date** – по дате, **rand** – в случайном порядке, **keywords** – статьи, похожие по названию для текущей страницы (должен быть подключен модуль «Поиск по сайту» и проиндексированы статьи);
- **images** – количество изображений, прикрепленных к статье;
- **images_variation** – тег размера изображений, задается в настройках модуля;
- **only_module** – выводить блок только на странице, к которой прикреплен модуль «Статьи»: **true** – выводить блок только на странице модуля, по умолчанию блок будет выводиться на всех страницах;
- **tag** – тег, прикрепленный к статьям;
- **defer** – маркер отложенной загрузки шаблонного тега: **event** – загрузка контента только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** – загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка контента только по желанию пользователя;

- **defer_title** – текстовая строка, выводимая на месте появления загружаемого контента с помощью отложенной загрузки шаблонного тега;
- **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

В шаблоне тега можно получить значение любого атрибута через переменную `$result["attributes"]["название атрибута"]`.

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;
- **element_type** – тип данных (*element* – элемент (по умолчанию), *cat* – категория);
- **sort** – сортировка комментариев: **date** – по дате (по умолчанию), **rand** – в случайном порядке;
- **defer** – маркер отложенной загрузки шаблонного тега: **event** – загрузка контента только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** – загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка контента только по желанию пользователя;

- **defer_title** – текстовая строка, выводимая на месте появления загружаемого контента с помощью отложенной загрузки шаблонного тега;
- **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

В шаблоне тега можно получить значение любого атрибута через переменную `$result["attributes"]["название атрибута"]`.

show_block module="consultant"

Подключает on-line консультант.

Атрибуты:

- **defer** – маркер отложенной загрузки шаблонного тега: **event** – загрузка контента только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** – загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка контента только по желанию пользователя;
- **defer_title** – текстовая строка, выводимая на месте появления загружаемого контента с помощью отложенной загрузки шаблонного тега;
- **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" system="jivosite">
```

выведет код для подключения on-line консультанта JivoSite

В шаблоне тега можно получить значение любого атрибута через переменную `$result["attributes"]["название атрибута"]`.

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. Можно указать отрицательное значение, тогда будут исключены вопросы из указанного раздела. По умолчанию выбираются все страницы;
- **cat_id** – категории вопросов, если в настройках модуля отмечено «Использовать категории». Идентификаторы категорий перечисляются через запятую. Можно указать отрицательное значение, тогда будут исключены вопросы из указанной категории. Можно указать значение **current**, тогда будут показаны вопросы из текущей (открытой) категории или из всех категорий, если ни одна категория не открыта. По умолчанию категория не учитывается, выводятся все вопросы;
- **sort** – сортировка вопросов: **date** – по дате (по умолчанию), **rand** – в случайном порядке;
- **often** – часто задаваемые вопросы : **true** – выводятся только вопросы с пометкой «Часто задаваемый вопрос», по умолчанию пометка «Часто задаваемый вопрос» игнорируется;
- **only_module** – выводить блок только на странице, к которой прикреплен модуль «Вопрос-Ответ»: **true** – выводить блок только на странице модуля, по умолчанию блок будет выводиться на всех страницах;
- **tag** – тег, прикрепленный к вопросам;
- **defer** – маркер отложенной загрузки шаблонного тега: **event** – загрузка контента только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** – загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка контента только по желанию пользователя;
- **defer_title** – текстовая строка, выводимая на месте появления загружаемого контента с помощью отложенной загрузки шаблонного тега;
- **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

В шаблоне тега можно получить значение любого атрибута через переменную `$result["attributes"]["название атрибута"]`.

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. Можно указать отрицательное значение, тогда будут исключены файлы из указанного раздела. По умолчанию выбираются все страницы;
- **cat_id** – категории файлов, если в настройках модуля отмечено «Использовать категории». Идентификаторы категорий перечисляются через запятую. Можно указать отрицательное значение, тогда будут исключены файлы из указанной категории. Можно указать значение **current**, тогда будут показаны файлы из текущей (открытой) категории или из всех категорий, если ни одна категория не открыта. По умолчанию категория не учитывается, выводятся все файлы;
- **sort** – сортировка файлов: по умолчанию как на странице модуля, **rand** – в случайном порядке, **date** – по дате;
- **images** – количество изображений, прикрепленных к файлу;
- **images_variation** – тег размера изображений, задается в настройках модуля;
- **only_module** – выводить блок только на странице, к которой прикреплен модуль «Файловый архив»: **true** – выводить блок только на странице модуля, по умолчанию блок будет выводиться на всех страницах;
- **tag** – тег, прикрепленный к файлам;
- **defer** – маркер отложенной загрузки шаблонного тега: **event** – загрузка контента только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** – загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка контента только по желанию пользователя;

- **defer_title** – текстовая строка, выводимая на месте появления загружаемого контента с помощью отложенной загрузки шаблонного тега;
- **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

В шаблоне тега можно получить значение любого атрибута через переменную `$result["attributes"]["название атрибута"]`.

show_block module="forum"

Выводит последние темы.

Атрибуты:

- **count** – количество выводимых тем (по умолчанию 3);
- **block_id** – блоки форума. Идентификаторы блоков перечисляются через запятую. Можно указать отрицательное значение, тогда будут исключены темы из блока. По умолчанию блок не учитывается, выводятся темы из всех блоков;
- **cat_id** – категории форума. Идентификаторы категорий перечисляются через запятую. Можно указать отрицательное значение, тогда будут исключены темы из указанной категории. По умолчанию категория не учитывается, выводятся темы из всех категорий;
- **sort** – сортировка тем: по умолчанию как на странице модуля, **date** – по дате, **rand** – в случайном порядке, **keywords** – темы, похожие по названию для текущей страницы;
- **only_module** – выводить блок только на странице, к которой прикреплен модуль «Форум»: **true** – выводить блок только на странице модуля, по умолчанию блок будет выводиться на всех страницах;
- **defer** – маркер отложенной загрузки шаблонного тега: **event** – загрузка контента только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** – загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка контента только по желанию пользователя;
- **defer_title** – текстовая строка, выводимая на месте появления загружаемого контента с помощью отложенной загрузки шаблонного тега;
- **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

В шаблоне тега можно получить значение любого атрибута через переменную `$result["attributes"]["название атрибута"]`.

show_block module="languages"

Выводит ссылки на альтернативные языковые версии сайта.

Атрибуты:

- **defer** – маркер отложенной загрузки шаблонного тега: **event** – загрузка контента только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** – загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка контента только по желанию пользователя;
- **defer_title** – текстовая строка, выводимая на месте появления загружаемого контента с помощью отложенной загрузки шаблонного тега;
- **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»

В шаблоне тега можно получить значение любого атрибута через переменную `$result["attributes"]["название атрибута"]`.

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_end_after_children_номер** – текст, выводимый после вложенных пунктов каждого 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_end_after_children_номер** – текст, выводимый после вложенных пунктов

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);

— **tag_active_child_end_after_children_номер** – текст, выводимый после вложенных пунктов 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;

— **count_level** – количество выводимых уровней меню, атрибут используется при оформлении меню атрибутами (template="");

— **defer** – маркер отложенной загрузки шаблонного тега: **event** – загрузка контента только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** – загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка контента только по желанию пользователя;

— **defer_title** – текстовая строка, выводимая на месте появления загружаемого контента с помощью отложенной загрузки шаблонного тега.

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>
```

В шаблоне тега можно получить значение любого атрибута через переменную `$result["attributes"]["название атрибута"]`.

show_block_messages module="forum"

Выводит блок сообщений.

Атрибуты:

- **count** – количество выводимых сообщений (по умолчанию 3);
- **block_id** – блоки форума. Идентификаторы блоков перечисляются через запятую. Можно указать отрицательное значение, тогда будут исключены сообщения из указанного блока. По умолчанию блок не учитывается, выводятся сообщения из всех блоков.;
- **cat_id** – категории форума. Идентификаторы категорий перечисляются через запятую. Можно указать отрицательное значение, тогда будут исключены сообщения из указанной категории. По умолчанию категория не учитывается, выводятся сообщения из всех категорий;
- **only_module** – выводить блок только на странице, к которой прикреплен модуль «Форум»: **true** – выводить блок только на странице модуля, по умолчанию блок будет выводиться на всех страницах;
- **defer** – маркер отложенной загрузки шаблонного тега: **event** – загрузка контента

только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** – загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка контента только по желанию пользователя;

— **defer_title** – текстовая строка, выводимая на месте появления загружаемого контента с помощью отложенной загрузки шаблонного тега;

— **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">
```

В шаблоне тега можно получить значение любого атрибута через переменную `$result["attributes"]["название атрибута"]`.

show_block module="mistakes"

Подключает модуль «Ошибка на сайте».

Атрибуты:

— **defer** – маркер отложенной загрузки шаблонного тега: **event** – загрузка контента только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** – загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка контента только по желанию пользователя;

— **defer_title** – текстовая строка, выводимая на месте появления загружаемого контента с помощью отложенной загрузки шаблонного тега.

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*.

В шаблоне тега можно получить значение любого атрибута через переменную `$result["attributes"]["название атрибута"]`.

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. Можно указать отрицательное значение, тогда будут исключены новости из указанного раздела. По умолчанию выбираются все страницы;
- **cat_id** – категории новостей, если в настройках модуля отмечено «Использовать категории». Идентификаторы категорий перечисляются через запятую. Можно указать отрицательное значение, тогда будут исключены новости из указанной категории. Можно указать значение **current**, тогда будут показаны новости из текущей (открытой) категории или из всех категорий, если ни одна категория не открыта. По умолчанию категория не учитывается, выводятся все новости;
- **sort** – сортировка новостей: **date** – по дате (по умолчанию), **keywords** – новости, похожие по названию для текущей страницы (должен быть подключен модуль «Поиск по сайту» и проиндексированы новости);
- **images** – количество изображений, прикрепленных к новости;
- **images_variation** – тег размера изображений, задается в настройках модуля;
- **only_module** – выводить блок только на странице, к которой прикреплен модуль «Новости»: **true** – выводить блок только на странице модуля, по умолчанию блок будет выводиться на всех страницах;
- **tag** – тег, прикрепленный к новостям;
- **defer** – маркер отложенной загрузки шаблонного тега: **event** – загрузка контента только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** – загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка контента только по желанию пользователя;
- **defer_title** – текстовая строка, выводимая на месте появления загружаемого контента с помощью отложенной загрузки шаблонного тега;
- **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

В шаблоне тега можно получить значение любого атрибута через переменную `$result["attributes"]["название атрибута"]`.

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;
- **defer** – маркер отложенной загрузки шаблонного тега: **event** – загрузка контента только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** – загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка контента только по желанию пользователя;
- **defer_title** – текстовая строка, выводимая на месте появления загружаемого контента с помощью отложенной загрузки шаблонного тега;
- **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

В шаблоне тега можно получить значение любого атрибута через переменную `$result["attributes"]["название атрибута"]`.

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. Можно указать отрицательное значение, тогда будут исключены фотографии из указанного раздела. По умолчанию выбираются все страницы;
- **cat_id** – альбомы фотографий, если в настройках модуля отмечено «Использовать альбомы». Идентификаторы альбомов перечисляются через запятую. Можно указать отрицательное значение, тогда будут исключены фотографии из указанной категории. Можно указать значение **current**, тогда будут показаны фотографии из текущей (открытой) категории или из всех категорий, если ни одна категория не открыта. 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;
- **only_module** – выводить блок только на странице, к которой прикреплен модуль «Фотогалерея»: **true** – выводить блок только на странице модуля, по умолчанию блок будет выводиться на всех страницах;
- **tag** – тег, прикрепленный к фотографиям;
- **defer** – маркер отложенной загрузки шаблонного тега: **event** – загрузка контента только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** – загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка контента только по желанию пользователя;
- **defer_title** – текстовая строка, выводимая на месте появления загружаемого контента с помощью отложенной загрузки шаблонного тега;
- **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

В шаблоне тега можно получить значение любого атрибута через переменную `$result["attributes"]["название атрибута"]`.

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 объявлению;
- **images_variation** – тег размера изображений, задается в настройках модуля;
- **defer** – маркер отложенной загрузки шаблонного тега: **event** – загрузка контента только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** – загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка контента только по желанию пользователя;
- **defer_title** – текстовая строка, выводимая на месте появления загружаемого контента с помощью отложенной загрузки шаблонного тега;
- **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

В шаблоне тега можно получить значение любого атрибута через переменную `$result["attributes"]["название атрибута"]`.

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;
- **defer** – маркер отложенной загрузки шаблонного тега: **event** – загрузка контента только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** – загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка контента только по желанию пользователя;
- **defer_title** – текстовая строка, выводимая на месте появления загружаемого контента с помощью отложенной загрузки шаблонного тега;
- **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

В шаблоне тега можно получить значение любого атрибута через переменную `$result["attributes"]["название атрибута"]`.

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);
- **defer** – маркер отложенной загрузки шаблонного тега: **event** – загрузка контента только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** – загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка контента только по желанию пользователя;

- **defer_title** – текстовая строка, выводимая на месте появления загружаемого контента с помощью отложенной загрузки шаблонного тега;
- **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

В шаблоне тега можно получить значение любого атрибута через переменную `$result["attributes"]["название атрибута"]`.

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;
- **defer** – маркер отложенной загрузки шаблонного тега: **event** – загрузка контента только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** – загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка контента только по желанию пользователя;
- **defer_title** – текстовая строка, выводимая на месте появления загружаемого контента с помощью отложенной загрузки шаблонного тега;
- **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

В шаблоне тега можно получить значение любого атрибута через переменную `$result["attributes"]["название атрибута"]`.

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;
- **defer** – маркер отложенной загрузки шаблонного тега: **event** – загрузка контента только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** – загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка контента только по желанию пользователя;
- **defer_title** – текстовая строка, выводимая на месте появления загружаемого контента с помощью отложенной загрузки шаблонного тега;
- **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

В шаблоне тега можно получить значение любого атрибута через переменную `$result["attributes"]["название атрибута"]`.

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;
- **defer** – маркер отложенной загрузки шаблонного тега: **event** – загрузка контента только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** – загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка контента только по желанию пользователя;
- **defer_title** – текстовая строка, выводимая на месте появления загружаемого контента с помощью отложенной загрузки шаблонного тега;
- **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

В шаблоне тега можно получить значение любого атрибута через переменную `$result["attributes"]["название атрибута"]`.

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;
- **defer** – маркер отложенной загрузки шаблонного тега: **event** – загрузка контента только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** – загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка контента только по желанию пользователя;
- **defer_title** – текстовая строка, выводимая на месте появления загружаемого контента с помощью отложенной загрузки шаблонного тега;

— **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

В шаблоне тега можно получить значение любого атрибута через переменную `$result["attributes"]["название атрибута"]`.

show_block module="reviews"

Выводит последние отзывы.

Атрибуты:

- **count** – количество выводимых отзывов (по умолчанию 3);
- **modules** – модули (разделенные запятой), к которым подключены отзывы;
- **element_id** – идентификаторы элементов модулей (разделенные запятой), к которым подключены отзывы;
- **element_type** – тип данных (*element* – элемент (по умолчанию), *cat* – категория, *brand* – производитель);
- **sort** – сортировка отзывов: **date** – по дате (по умолчанию), **rand** – в случайном порядке;
- **defer** – маркер отложенной загрузки шаблонного тега: **event** – загрузка контента только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** – загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка контента только по желанию пользователя;
- **defer_title** – текстовая строка, выводимая на месте появления загружаемого контента с помощью отложенной загрузки шаблонного тега;
- **template** – шаблон тега (файл `modules/reviews/views/reviews.view.show_block_`**template**`.php`; по умолчанию шаблон `modules/reviews/views/reviews.view.show_block.php`).

Пример:

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

В шаблоне тега можно получить значение любого атрибута через переменную

`$result["attributes"]["название атрибута"]`.

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. Можно указать отрицательное значение, тогда будут исключены товары из указанного раздела. По умолчанию выбираются все страницы;
- **cat_id** – категории товаров, если в настройках модуля отмечено «Использовать категории». Можно указать отрицательное значение, тогда будут исключены товары из указанной категории. Идентификаторы категорий перечисляются через запятую. Можно указать значение **current**, тогда будут показаны товары из текущей (открытой) категории магазина или из всех категорий, если ни одна категория не открыта. По умолчанию категория не учитывается, выводятся все товары;
- **ids** – Товары. Можно указать отрицательное значение, тогда будут исключены товары. Идентификаторы товаров перечисляются через запятую. По умолчанию товары не учитываются, выводятся все товары;
- **brand_id** – производители товаров. Можно указать отрицательное значение, тогда будут исключены товары указанного производителя. Идентификаторы производителя перечисляются через запятую. По умолчанию производитель не учитывается, выводятся все товары;
- **sort** – сортировка товаров: по умолчанию как на странице модуля, **date** – по дате, **rand** – в случайном порядке, **price** – по цене, **sale** – по количеству продаж;
- **images** – количество изображений, прикрепленных к товару;
- **images_variation** – тег размера изображений, задается в настройках модуля;
- **param** – значения дополнительных характеристик;

Пример:

Товары обладают следующими характеристиками:

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

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

аналогов. Символы `<` и `>` нужно заменять HTML-мнемониками `<` и `>`;

```
<insert name="show_block" module="shop" param="3=5&3=6&10&12&16=0">
```

Номер (или идентификатор) характеристики можно посмотреть, если подвести курсор к названию характеристики в списке характеристик в административной части. Появится всплывающая подсказка «Редактировать (номер характеристики)».

Номер (или идентификатор) значения характеристики можно посмотреть, если при редактировании характеристики подвести курсора на нужное значение. Появится всплывающая подсказка «ID: номер».

- **hits_only** – выводить только товары с пометкой «Хит»: **true** – выводить только товары с пометкой «Хит», по умолчанию пометка «Хит» будет игнорироваться;
- **action_only** – выводить только товары с пометкой «Акция»: **true** – выводить только товары с пометкой «Акция», по умолчанию пометка «Акция» будет игнорироваться;
- **new_only** – выводить только товары с пометкой «Новинка»: **true** – выводить только товары с пометкой «Новинка», по умолчанию пометка «Новинка» будет игнорироваться;
- **discount_only** – выводить только товары, на которые действует скидка: **true** – выводить только товары, на которые действует скидка, по умолчанию скидка у товаров игнорируется;
- **only_module** – выводить блок только на странице, к которой прикреплен модуль «Магазин»: **true** – выводить блок только на странице модуля, по умолчанию блок будет выводиться на всех страницах;
- **tag** – тег, прикрепленный к товарам;
- **defer** – маркер отложенной загрузки шаблонного тега: **event** – загрузка контента только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** – загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка контента только по желанию пользователя;
- **defer_title** – текстовая строка, выводимая на месте появления загружаемого контента с помощью отложенной загрузки шаблонного тега;
- **template** – шаблон тега (файл `modules/shop/views/shop.view.show_block_template.php`; по умолчанию шаблон `modules/shop/views/shop.view.show_block.php`).

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

В шаблоне тега можно получить значение любого атрибута через переменную `$result["attributes"]["название атрибута"]`.

show_block module="site"

Выводит содержимое блока на сайте, номер которой передан в виде атрибута id.

Атрибуты:

- **id** – идентификатор блока;
- **defer** – маркер отложенной загрузки шаблонного тега: **event** – загрузка контента только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** – загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка контента только по желанию пользователя;
- **defer_title** – текстовая строка, выводимая на месте появления загружаемого контента с помощью отложенной загрузки шаблонного тега;
- **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»

В шаблоне тега можно получить значение любого атрибута через переменную `$result["attributes"]["название атрибута"]`.

show_block module="tags"

Выводит список тегов.

Атрибуты:

- **defer** – маркер отложенной загрузки шаблонного тега: **event** – загрузка контента только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** – загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка контента только по желанию пользователя;
- **defer_title** – текстовая строка, выводимая на месте появления загружаемого контента с помощью отложенной загрузки шаблонного тега;
- **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

В шаблоне тега можно получить значение любого атрибута через переменную `$result["attributes"]["название атрибута"]`.

show_block module="users"

Выводит статистику пользователей на сайте.

Атрибуты:

- **defer** – маркер отложенной загрузки шаблонного тега: **event** – загрузка контента только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** – загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка контента только по желанию пользователя;
- **defer_title** – текстовая строка, выводимая на месте появления загружаемого контента с помощью отложенной загрузки шаблонного тега;
- **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.»

В шаблоне тега можно получить значение любого атрибута через переменную `$result["attributes"]["название атрибута"]`.

show_block module="votes"

Выводит опросы.

Атрибуты:

- **id** – задает номер вопроса;
- **count** – количество вопросов. Значение all выведет все вопросы. По умолчанию 1;
- **sort** – сортировка опросов: по умолчанию ручная сортировка как в административной части, **rand** – в случайном порядке;
- **defer** – маркер отложенной загрузки шаблонного тега: **event** – загрузка контента только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** – загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка контента только по желанию пользователя;
- **defer_title** – текстовая строка, выводимая на месте появления загружаемого контента с помощью отложенной загрузки шаблонного тега;
- **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

В шаблоне тега можно получить значение любого атрибута через переменную `$result["attributes"]["название атрибута"]`.

show_block module="wishlist"

Выводит информацию о товарах в списке пожеланий.

Атрибуты:

— **defer** – маркер отложенной загрузки шаблонного тега: **event** – загрузка контента только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** – загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка контента только по желанию пользователя;

— **defer_title** – текстовая строка, выводимая на месте появления загружаемого контента с помощью отложенной загрузки шаблонного тега;

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

Пример:

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

В шаблоне тега можно получить значение любого атрибута через переменную `$result["attributes"]["название атрибута"]`.

show_body

Выводит основной контент страницы: заголовка (если не запрещен его вывод в настройке странице «Не показывать заголовков»), текста страницы и прикрепленного модуля.

Заменяет три тега: `show_h1`, `show_text`, `show_module`.

Пример:

```
<insert name="show_body">
```

Шаблон в файле *themes/functions/show_body.php*.

В шаблоне тега можно получить значение любого атрибута через переменную `$attributes["название атрибута"]`.

show_brand module="shop"

Выводит несколько производителей.

Атрибуты:

— **count** – количество выводимых производителей (по умолчанию выводятся все производители);

— **site_id** – страницы, к которым прикреплен модуль. Идентификаторы страниц перечисляются через запятую. Можно указать отрицательное значение, тогда будут исключены производители из указанного раздела. По умолчанию выбираются все страницы;

- **cat_id** – категории товаров, если в настройках модуля отмечено «Использовать категории». Можно указать отрицательное значение, тогда будут исключены производители из указанной категории. Идентификаторы категорий перечисляются через запятую. Можно указать значение **current**, тогда будут показаны производители из текущей (открытой) категории магазина или из всех категорий, если ни одна категория не открыта. По умолчанию категория не учитывается, выводятся все производители;
- **sort** – сортировка производителей: по умолчанию как на странице модуля, **name** – по имени, **rand** – в случайном порядке;
- **images** – количество изображений, прикрепленных к производителю;
- **images_variation** – тег размера изображений, задается в настройках модуля;
- **only_module** – выводить блок только на странице, к которой прикреплен модуль «Магазин»: **true** – выводить блок только на странице модуля, по умолчанию блок будет выводиться на всех страницах;
- **defer** – маркер отложенной загрузки шаблонного тега: **event** – загрузка контента только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** – загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка контента только по желанию пользователя;
- **defer_title** – текстовая строка, выводимая на месте появления загружаемого контента с помощью отложенной загрузки шаблонного тега;
- **template** – шаблон тега (файл *modules/shop/views/shop.view.show_brand_template.php*; по умолчанию шаблон *modules/shop/views/shop.view.show_brand.php*).

Пример:

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

В шаблоне тега можно получить значение любого атрибута через переменную `$result["attributes"]["название атрибута"]`.

show_breadcrumb

Выводит навигацию по сайту «Хлебные крошки».

Атрибуты:

- **separator** – разделитель ссылок в навигации;
- **current** – выводить текущий пункт: **true** – выводить ссылку на текущую страницу, по умолчанию ссылка на текущую страницу не выводится.

Пример:

```
<insert name="show_breadcrumb">
```

Шаблон в файле *themes/functions/show_breadcrumb.php*.

В шаблоне тега можно получить значение любого атрибута через переменную `$attributes["название атрибута"]`.

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»;
- **detail** – детализация (**day** – дни, **month** – месяца (по умолчанию), **year** – годы);
- **only_module** – выводить блок только на странице «Новости»: **true** – выводить блок только на странице модуля, по умолчанию блок будет выводиться на всех страницах;
- **defer** – маркер отложенной загрузки шаблонного тега: **event** – загрузка контента только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** – загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка контента только по желанию пользователя;
- **defer_title** – текстовая строка, выводимая на месте появления загружаемого контента с помощью отложенной загрузки шаблонного тега;
- **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

В шаблоне тега можно получить значение любого атрибута через переменную `$result["attributes"]["название атрибута"]`.

show_category module="shop"

Выводит несколько категорий.

Атрибуты:

- **site_id** – страницы, к которым прикреплен модуль. Идентификаторы страниц перечисляются через запятую. Можно указать отрицательное значение, тогда будут исключены категории из указанного раздела. По умолчанию выбираются все страницы;
- **images** – количество изображений, прикрепленных к категории;
- **images_variation** – тег размера изображений, задается в настройках модуля;
- **only_module** – выводить блок только на странице, к которой прикреплен модуль «Магазин»: **true** – выводить блок только на странице модуля, по умолчанию блок будет выводиться на всех страницах;
- **count_level** – количество уровней;
- **number_elements** – выводить количество товаров в категории: **true** – выводить количество товаров, по умолчанию количество не выводится;
- **defer** – маркер отложенной загрузки шаблонного тега: **event** – загрузка контента только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** – загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка контента только по желанию пользователя;
- **defer_title** – текстовая строка, выводимая на месте появления загружаемого контента с помощью отложенной загрузки шаблонного тега;
- **template** – шаблон тега (файл `modules/shop/views/shop.view.show_category_**template.php`; по умолчанию шаблон `modules/shop/views/shop.view.show_category.php`).

Пример:

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

В шаблоне тега можно получить значение любого атрибута через переменную `$result["attributes"]["название атрибута"]`.

show_comments module="site"

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

Атрибуты:

- **defer** – маркер отложенной загрузки шаблонного тега: **event** – загрузка контента только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** – загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка контента только по желанию пользователя;
- **defer_title** – текстовая строка, выводимая на месте появления загружаемого контента с помощью отложенной загрузки шаблонного тега;
- **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

В шаблоне тега можно получить значение любого атрибута через переменную `$result["attributes"]["название атрибута"]`.

show_css

Подключает CSS-файлы. При включенном режиме разработки файлы будут объединены и сжаты, что приведет к более быстрой загрузке файлов. Если существуют какие-то проблемы при включенном сжатии, подключите CSS-файлы стандартным HTML-тегом `<link rel="stylesheet" type="text/css" ...>`.

Атрибуты:

- **files** – перечень CSS-файлов, которые нужно подключить. Файлы должны размещаться в папке `css`. Если файлов несколько, то названия должны быть разделены запятыми.

Пример:

```
<insert name="show_css" files="default.css, style.css">
```

Тег подключит файлы `css/default.css`, `css/style.css`. Если файлы находятся в кастомной теме, то пути к ним корректно пропишутся. Файлы будут объединены, сжаты и станут

быстрее загружаться.

Шаблон в файле `themes/functions/show_css.php`.

В шаблоне тега можно получить значение любого атрибута через переменную `$attributes["название атрибута"]`.

custom

Выводит путь до файла с учетом кастомизации.

Атрибуты:

— **path** – исходный путь до файла;

Пример:

```
<insert name="custom" path="img/logo.png">
```

Выведет:

- `custom/my/img/logo.png` — если файл найден в текущей теме (в примере тема называется `my`);
- `img/logo.png` — если файла нет в текущей теме, но он найден в корне сайта;
- пустое значение — если файла нет нигде.

— **absolute** – путь абсолютный: **true** – тег выведет полный путь до файла, по умолчанию тег выведет относительный путь до файла без доменного имени;

— **compress** – сжатие файла: **js** - тип js, **css** - тип css.

Пример:

```
<insert name="custom" path="img/logo.png">
```

Выведет `custom/my/img/logo.png`.

```
<insert name="custom" path="img/logo.png" absolute="true">
```

Выведет `https://site.ru/custom/my/img/logo.png`.

Шаблон в файле `themes/functions/custom.php`.

В шаблоне тега можно получить значение любого атрибута через переменную `$attributes["название атрибута"]`.

D

show_description

Выводит мета-тег description страницы.

Пример:

```
<meta name="description" content="<insert name="show_description">">
```

выведет описание текущей страницы

Шаблон в файле *themes/functions/show_description.php*.

В шаблоне тега можно получить значение любого атрибута через переменную `$attributes["название атрибута"]`.

show_dynamic module="site"

Выводит содержимое динамического блока, номер которой передан в виде атрибута id.

Атрибуты:

- **id** – идентификатор динамического блока;
- **element_id** – номер элемента, для которого будет выведено значение блока, по умолчанию текущий элемент;
- **module_name** – модуль элемента, для которого будет выведено значение блока, по умолчанию текущий модуль;
- **element_type** – тип элемента, для которого будет выведено значение блока, по умолчанию тип текущего элемента;
- **defer** – маркер отложенной загрузки шаблонного тега: **event** – загрузка контента только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** – загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка контента только по желанию пользователя;
- **defer_title** – текстовая строка, выводимая на месте появления загружаемого контента с помощью отложенной загрузки шаблонного тега;
- **template** – шаблон тега (файл *modules/site/views/site.view.show_dynamic_template.php*; по умолчанию шаблон *modules/site/views/site.view.show_dynamic.php*).

Пример:

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

В шаблоне тега можно получить значение любого атрибута через переменную `$result["attributes"]["название атрибута"]`.

F

show_form module="ab"

Выводит форму добавления сообщения. Для правильной работы тега должна существовать страница, к которой прикреплен модуль Объявления.

Атрибуты:

- **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»;
- **only_module** – выводить форму только на странице, к которой прикреплен модуль «Объявления»: **true** – выводить форму только на странице модуля, по умолчанию форма будет выводиться на всех страницах;
- **defer** – маркер отложенной загрузки шаблонного тега: **event** – загрузка контента только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** – загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка контента только по желанию пользователя;
- **defer_title** – текстовая строка, выводимая на месте появления загружаемого контента с помощью отложенной загрузки шаблонного тега;
- **template** – шаблон тега (файл `modules/ab/views/ab.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

В шаблоне тега можно получить значение любого атрибута через переменную `$result["attributes"]["название атрибута"]`.

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»;
- **only_module** – выводить форму только на странице, к которой прикреплен модуль «Вопрос-Ответ»: **true** – выводить форму только на странице модуля, по умолчанию форма будет выводиться на всех страницах;
- **defer** – маркер отложенной загрузки шаблонного тега: **event** – загрузка контента только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** – загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка контента только по желанию пользователя;
- **defer_title** – текстовая строка, выводимая на месте появления загружаемого контента с помощью отложенной загрузки шаблонного тега;
- **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

В шаблоне тега можно получить значение любого атрибута через переменную `$result["attributes"]["название атрибута"]`.

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;
- **defer** – маркер отложенной загрузки шаблонного тега: **event** – загрузка контента только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** – загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка контента только по желанию пользователя;
- **defer_title** – текстовая строка, выводимая на месте появления загружаемого контента с помощью отложенной загрузки шаблонного тега;
- **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

В шаблоне тега можно получить значение любого атрибута через переменную `$result["attributes"]["название атрибута"]`.

show_form module="subscription"

Выводит форму подписки на рассылки.

Атрибуты:

- **defer** – маркер отложенной загрузки шаблонного тега: **event** – загрузка контента только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** – загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка контента только по желанию пользователя;
- **defer_title** – текстовая строка, выводимая на месте появления загружаемого контента с помощью отложенной загрузки шаблонного тега;
- **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">
```

В шаблоне тега можно получить значение любого атрибута через переменную `$result["attributes"]["название атрибута"]`.

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 в файле `themes/functions/show_h1.php`.

В шаблоне тега можно получить значение любого атрибута через переменную `$attributes["название атрибута"]`.

show_head

Формирует часть HTML-шапки сайта. Включает в себя в том числе теги: `show_title`, `show_description`, `show_keywords`.

Пример:

```
<insert name="show_head">
```

Шаблон в файле `themes/functions/show_head.php`.

В шаблоне тега можно получить значение любого атрибута через переменную `$attributes["название атрибута"]`.

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 ``, но ликвидирует ссылки, которые никуда не ведут.

Атрибуты:

- **rewrite** – псевдоссылка страницы, на которую ведет ссылка, например: `news`;
- **img** – адрес изображения, использующегося в качестве ссылки, например:

img/logo.png, можно добавлять `_LANG`, чтобы подставить ID текущего языка;

— **img_act** – адрес изображения на текущей странице, например: `img/home_act.gif`, можно добавлять `_LANG`, чтобы подставить ID текущего языка;

— **width** – ширина изображения;

— **height** – высота изображения;

— **class** – класс для ссылки;

— **alt** – альтернативный тег для изображения или текст ссылки. Если задано **title**, будет подставлено название сайта из параметров сайта. Если задано **url**, будет подставлена ссылка на главную страницу сайта.

Пример:

```
<insert name="show_href" rewrite="shop/cart" alt="Корзина">
```

Выведет:

- на странице "Корзина"

[Корзина](#)

- на остальных страницах

```
<a href="http://site.ru/shop/cart/" title="Корзина">Корзина</a>
```

- на остальных страницах английской версии, если слово "Корзина" переведено в интерфейсе "Языки сайта"

```
<a href="http://site.ru/eng/shop/cart/" title="Cart">Cart</a>
```

```
<insert name="show_href" alt="url" class="logo">
```

Выведет:

- на главной странице

<http://site.ru/>

- на внутренней странице

```
<a href="http://site.ru/" title="http://site.ru/" class="logo">http://site.ru/</a>
```

- на внутренней странице английской версии

```
<a href="http://site.ru/eng/" title="http://site.ru/" class="logo">http://site.ru/</a>
```

```
<insert name="show_href" img="img/logo_LANG.png" alt="title" width="220" height="80">
```

Выведет:

- на главной странице

```

```

- на внутренней странице

```
<a href="http://site.ru/" title="Название сайта"></a>
```

- на внутренней странице английской версии

```
<a href="http://site.ru/eng/" title="Site name"></a>
```

```
<insert name="show_href" img="img/map.png" img_act="img/map_act.png" alt="Карта сайта" rewrite="map" width="10" height="20">
```

Выведет:

- на странице "Карта сайта"

```

```

- на всех остальных страницах

```
<a href="http://site.ru/map/" title="Карта сайта"></a>
```

Шаблон в файле *themes/functions/show_href.php*.

В шаблоне тега можно получить значение любого атрибута через переменную `$attributes["название атрибута"]`.

I

show_images module="site"

Выводит изображения, прикрепленные к странице сайта, если в конфигурации модуля «Страницы сайта» включены параметры «Использовать изображения».

Атрибуты:

- **defer** – маркер отложенной загрузки шаблонного тега: **event** – загрузка контента только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** – загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка контента только по желанию пользователя;
- **defer_title** – текстовая строка, выводимая на месте появления загружаемого контента с помощью отложенной загрузки шаблонного тега;
- **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 группу фотографий, прикрепленных к текущей странице

В шаблоне тега можно получить значение любого атрибута через переменную `$result["attributes"][$"название атрибута"]`.

show_include

Подключает файл-блок шаблона.

Атрибуты:

- **file** – имя PHP-файла из папки *themes/blocks* без расширения.

Пример:

```
<insert name="show_include" file="diafan">
```

подключит файл *themes/blocks/diafan.php*

Шаблон в файле *themes/functions/show_include.php*.

В шаблоне тега можно получить значение любого атрибута через переменную `$attributes["название атрибута"]`.

J

show_js

Подключает JS-файлы. Тег нужно добавить перед `</body>`.

Пример:

```
<html>
<head>...</head>
<body>
....
<insert name="show_js">
</body>
</html>
```

Шаблон в файле *themes/functions/show_js.php*.

В шаблоне тега можно получить значение любого атрибута через переменную `$attributes["название атрибута"]`.

K

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 в файле *themes/functions/show_keywords.php*.

В шаблоне тега можно получить значение любого атрибута через переменную `$attributes["название атрибута"]`.

L

language

Выводит номер текущей языковой версии сайта.

Пример:

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

Шаблон в файле *themes/functions/language.php*.

В шаблоне тега можно получить значение любого атрибута через переменную `$attributes["название атрибута"]`.

show_last_order module="cart"

Выводит информацию о последнем совершенном заказе.

Атрибуты:

- **defer** – маркер отложенной загрузки шаблонного тега: **event** – загрузка контента только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** – загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка контента только по желанию пользователя;
- **defer_title** – текстовая строка, выводимая на месте появления загружаемого контента с помощью отложенной загрузки шаблонного тега;
- **template** – шаблон тега (файл *modules/cart/views/cart.view.show_last_order_template.php*; по умолчанию шаблон *modules/cart/views/cart.view.show_last_order.php*).

Пример:

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

В шаблоне тега можно получить значение любого атрибута через переменную `$result["attributes"]["название атрибута"]`.

show_links module="site"

Выводит ссылки на страницы нижнего уровня, принадлежащие текущей странице.

Атрибуты:

- **defer** – маркер отложенной загрузки шаблонного тега: **event** – загрузка контента только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** – загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка контента только по желанию пользователя;
- **defer_title** – текстовая строка, выводимая на месте появления загружаемого контента с помощью отложенной загрузки шаблонного тега;

— **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">
```

В шаблоне тега можно получить значение любого атрибута через переменную `$result["attributes"]["название атрибута"]`.

show_login module="registration"

Выводит форму авторизации или приветствие и ссылки на редактирование данных и выход, если пользователь авторизован.

Атрибуты:

— **defer** – маркер отложенной загрузки шаблонного тега: **event** – загрузка контента только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** – загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка контента только по желанию пользователя;

— **defer_title** – текстовая строка, выводимая на месте появления загружаемого контента с помощью отложенной загрузки шаблонного тега;

— **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

В шаблоне тега можно получить значение любого атрибута через переменную `$result["attributes"]["название атрибута"]`.

М

module

Выводит подключенный к странице модуль.

Пример:

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

Шаблон в файле *themes/functions/module.php*.

В шаблоне тега можно получить значение любого атрибута через переменную `$attributes["название атрибута"]`.

show_module

Выводит контент модуля.

Пример:

```
<insert name="show_module">
```

Шаблон в файле *themes/functions/show_module.php*.

В шаблоне тега можно получить значение любого атрибута через переменную `$attributes["название атрибута"]`.

P

path

Выводит адрес сайта.

Пример:

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

Выведет *http://site.ru/*.

Шаблон в файле *themes/functions/path.php*.

В шаблоне тега можно получить значение любого атрибута через переменную `$attributes["название атрибута"]`.

path_url

Выводит адрес сайта, с учетом языковой версии сайта.

Атрибуты:

- **mobile** – признак мобильной версии: yes – в адресе будет включено "m/", если страница – мобильная версия (по умолчанию); no – в адресе будет исключено "m/" даже если страница – мобильная версия.

Пример:

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

Выведет:

- *http://site.ru/* для русской версии;
- *http://site.ru/eng/* для английской версии;

Шаблон в файле *themes/functions/path_url.php*.

В шаблоне тега можно получить значение любого атрибута через переменную `$attributes["название атрибута"]`.

show_previous_next module="site"

Выводит ссылки на предыдущую и последующую страницы.

Атрибуты:

- **defer** – маркер отложенной загрузки шаблонного тега; **event** – загрузка контента только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** – загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка контента только по желанию пользователя;
- **defer_title** – текстовая строка, выводимая на месте появления загружаемого контента с помощью отложенной загрузки шаблонного тега;
- **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">
```

В шаблоне тега можно получить значение любого атрибута через переменную `$result["attributes"]["название атрибута"]`.

show_protect

Выводит JS код запрещающий копирование контента на сайте.

Пример:

```
<body<insert name="show_protect">>
```

Шаблон в файле *themes/functions/show_protect.php*.

В шаблоне тега можно получить значение любого атрибута через переменную `$attributes["название атрибута"]`.

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:

- **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. Можно указать отрицательное значение, тогда указанные страницы будут исключены из списка;
- **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 отрицательное значение, тогда указанные категории будут исключены из списка. Можно указать значение **all**, тогда поиск будет осуществляться по всем категориям объявлений и в форме будут участвовать только общие характеристики. Атрибут не обязателен;
- **ajax** – подгружать результаты поиска без перезагрузки страницы. Результаты подгружаются только если открыта страница со списком объявлений, иначе поиск работает обычным образом: **true** – результаты поиска подгружаются, по умолчанию будет перезагружена вся страница;
- **only_module** – выводить блок только на странице, к которой прикреплен модуль «Объявления»: **true** – выводить блок только на странице модуля, по умолчанию блок будет выводиться на всех страницах;
- **defer** – маркер отложенной загрузки шаблонного тега: **event** – загрузка контента только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** – загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка контента только по желанию пользователя;

- **defer_title** – текстовая строка, выводимая на месте появления загружаемого контента с помощью отложенной загрузки шаблонного тега;
- **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

В шаблоне тега можно получить значение любого атрибута через переменную `$result["attributes"]["название атрибута"]`.

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»;
- **ajax** – подгружать результаты поиска без перезагрузки страницы.: **true** – результаты поиска подгружаются, по умолчанию будет перезагружена вся страница.;
- **defer** – маркер отложенной загрузки шаблонного тега: **event** – загрузка контента только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** – загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка контента только по желанию пользователя;
- **defer_title** – текстовая строка, выводимая на месте появления загружаемого контента с помощью отложенной загрузки шаблонного тега;
- **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

В шаблоне тега можно получить значение любого атрибута через переменную

`$result["attributes"] ["название атрибута"]`.

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 артикулу», «Искать товары по акции», «Искать по новинкам», «Искать по хитам»). 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:

- **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. Можно указать отрицательное значение, тогда указанные страницы будут исключены из списка;
- **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 отрицательное значение, тогда указанные категории будут исключены из списка. Можно указать значение **all**, тогда поиск будет осуществляться по всем категориям товаров и в форме будут участвовать только общие характеристики. Атрибут не обязателен;
- **ajax** – подгружать результаты поиска без перезагрузки страницы: **true** – результаты поиска подгружаются, по умолчанию будет перезагружена вся страница. Результаты подгружаются только если открыта страница со списком товаров, иначе поиск работает обычным образом;
- **only_module** – выводить форму поиска только на странице модуля «Магазин»: **true** – выводить форму только на странице модуля, по умолчанию форма будет выводиться на всех страницах;
- **defer** – маркер отложенной загрузки шаблонного тега: **event** – загрузка контента только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** – загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка контента только по желанию пользователя;
- **defer_title** – текстовая строка, выводимая на месте появления загружаемого

контента с помощью отложенной загрузки шаблонного тега;

— **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

В шаблоне тега можно получить значение любого атрибута через переменную `$result["attributes"]["название атрибута"]`.

show module="reviews"

Выводит отзывы и форму добавления отзывов для текущей страницы.

Атрибуты:

- **modules** – модули (разделенные запятой), к которым подключены отзывы;
- **element_types** – типы данных (разделенные запятой): *element* – элемент (по умолчанию), *cat* – категория, *brand* – производитель. Если не задан, то подключается ко всем типам данных модуля;
- **defer** – маркер отложенной загрузки шаблонного тега: **event** – загрузка контента только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** – загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка контента только по желанию пользователя;
- **defer_title** – текстовая строка, выводимая на месте появления загружаемого контента с помощью отложенной загрузки шаблонного тега;
- **template** – шаблон тега (файл `modules/reviews/views/reviews.view.show_**template.php`; по умолчанию шаблон `modules/reviews/views/reviews.view.show.php`).

Пример:

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

В шаблоне тега можно получить значение любого атрибута через переменную `$result["attributes"]["название атрибута"]`.

show_social_links

Выводит ссылки на социальные сети.

Пример:

```
<insert name="show_social_links">
```

Шаблон в файле *themes/functions/show_social_links.php*.

В шаблоне тега можно получить значение любого атрибута через переменную `$attributes["название атрибута"]`.

T

show_tags module="site"

Выводит теги (слова-якоря), прикрепленные к странице сайта, если в конфигурации модуля «Страницы сайты» подключены теги.

Атрибуты:

- **defer** – маркер отложенной загрузки шаблонного тега: **event** – загрузка контента только по желанию пользователя при нажатии кнопки "Загрузить", **emergence** – загрузка контента только при появлении в окне браузера клиента, **async** – асинхронная (одновременная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, **sync** – синхронная (последовательная) загрузка контента совместно с контентом шаблонных тегов с тем же маркером, по умолчанию загрузка контента только по желанию пользователя;
- **defer_title** – текстовая строка, выводимая на месте появления загружаемого контента с помощью отложенной загрузки шаблонного тега;
- **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

В шаблоне тега можно получить значение любого атрибута через переменную `$result["attributes"]["название атрибута"]`.

show_text

Выводит текст страницы.

Пример:

```
<insert name="show_text">
```

Шаблон в файле *themes/functions/show_text.php*.

В шаблоне тега можно получить значение любого атрибута через переменную `$attributes["название атрибута"]`.

show_title

Выводит заголовок. Используется между тегами `<title></title>` в шапке сайта.

Пример:

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

выведет заголовок текущей страницы

Шаблон в файле *themes/functions/show_title.php*.

В шаблоне тега можно получить значение любого атрибута через переменную `$attributes["название атрибута"]`.

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 в файле *themes/functions/show_year.php*.

В шаблоне тега можно получить значение любого атрибута через переменную `$attributes["название атрибута"]`.