

Система управления сайтами NetCat
версия 4.5

Руководство по модулям

Компания «НетКэт»
Москва, 2011 г.

Внимание! Право на тиражирование программных компонентов и документации принадлежит компании «НетКэт». Приобретая систему NetCat, вы автоматически соглашаетесь не допускать копирование программ и документации без письменного разрешения компании «НетКэт», за исключением копирования документации в электронном виде.

© 1999-2011 ООО «НетКэт»

Компания «НетКэт», отдел разработки
Телефон: (495) 783-60-21 (многоканальный)
Сайт: <http://www.netcat.ru>
Электронная почта: support@netcat.ru

Разработка системы управления сайтами NetCat: К. Хряпин, В. Островский, В. Бахренков,
Р. Сакал, Н. Громин, Д. Спирин, А.Наливайко, Д.Варин
Документация: Н. Громин, Д. Спирин, Д. Васильев, В. Бахренков, Д.Варин
Руководство по модулям версии 4.5.

Оглавление

ГОЛОСОВАНИЕ.....	10
ПОИСК ПО САЙТУ.....	12
Обзор возможностей модуля.....	12
Поиск на сайте:.....	12
Индексирование документов на сайте:.....	12
Сбор информации для администратора сайта:.....	13
Оптимизация сайта для внешних поисковых систем:.....	13
Язык запросов.....	13
Интерфейс модуля в панели управления сайтом.....	15
Общие.....	15
Индексирование.....	15
Списки.....	16
Области индексирования.....	17
Задание списка страниц, подлежащих индексированию.....	17
Указание областей HTML-страниц, подлежащих индексированию.....	19
Сохранение частей страниц для использования в результатах поиска.....	20
Области поиска на сайте.....	22
Запуск по расписанию. Постановка заданий в очередь.....	22
Настройка для запуска индексирования по расписанию через cron.....	23
Настройка для запуска индексирования по расписанию через «Управление задачами».....	23
Правила индексирования.....	23
Постановка задачи переиндексирования в очередь.....	24
Интеграция модуля в макеты дизайна сайта.....	25
Простая форма поиска.....	25
Расширенная форма поиска.....	27
Вывод результатов поиска.....	27
Стилизация списка подсказок.....	29
Разработка расширений модуля.....	30
Обзор архитектуры модуля.....	30
Обработчики документов различных типов.....	32
Анализаторы текста.....	34
Текстовые фильтры.....	35
Корректировщики запросов.....	35
Подключение других поисковых систем.....	36
СТАТИСТИКА ПОСЕЩЕНИЙ.....	39
Установка.....	39
Настройка.....	39
УПРАВЛЕНИЕ ПОДПИСКАМИ И РАССЫЛКАМИ.....	42
Общее описание.....	42
Установка.....	42
Требования.....	42
Типы рассылок.....	42
Подписка на раздел.....	43
Регулярная рассылка.....	44
Сервисная рассылка.....	44
Управление рассылками и подписчиками.....	44
Список рассылок.....	44
Создание и изменение рассылки.....	45
Список подписчиков.....	46

Подписки пользователя.....	46
СТАТИСТИКА РАССЫЛОК.....	46
ЕДИНОРАЗОВАЯ РАССЫЛКА.....	47
НАСТРОЙКИ МОДУЛЯ.....	47
ДОСТУПЫ НА РАССЫЛКУ И НАЗНАЧЕНИЕ ПРАВ.....	49
ПРИНЦИП ФОРМИРОВАНИЯ ПИСЕМ.....	49
ШАБЛОНЫ ПИСЕМ, УСЛОВИЯ И ОТПРАВКА ПИСЕМ.....	50
<i>Шаблоны писем.....</i>	50
<i>Условие подписки.....</i>	50
<i>Условия рассылок и отправка писем.....</i>	50
СПИСОК ДОСТУПНЫХ ПЕРЕМЕННЫХ И МАКРОПЕРЕМЕННЫХ.....	51
<i>Условие рассылки.....</i>	51
<i>Действие после подписки.....</i>	52
<i>Тема письма.....</i>	52
<i>Хэдер и футер.....</i>	53
<i>Основная часть.....</i>	53
<i>Подтверждение подписки.....</i>	54
<i>Служебные переменные.....</i>	54
ПЕРИОДЫ ПОЛУЧЕНИЯ ПИСЕМ.....	54
СПОСОБЫ ПОДПИСКИ.....	54
СТРУКТУРА ТАБЛИЦ.....	56
КОМПОНЕНТ «СПИСОК ПОДПИСОК».....	56
ПОДПИСКА НА ОБЪЕКТ.....	57
ИНТЕГРАЦИЯ С МОДУЛЕМ «ФОРУМ2».....	58
СОВМЕСТИМОСТЬ С ПРЕДЫДУЩИМИ ВЕРСИЯМИ.....	59
ОРГАНИЗАЦИЯ ПОДПИСКИ ДЛЯ НЕЗАРЕГИСТРИРОВАННЫХ ПОЛЬЗОВАТЕЛЕЙ.....	59
КОНСТАНТЫ МОДУЛЯ.....	61
КЛАССЫ МОДУЛЯ.....	61
ЛИЧНЫЙ КАБИНЕТ.....	68
УСТАНОВКА И ТРЕБОВАНИЯ.....	68
РЕГИСТРАЦИЯ.....	69
<i>Форма регистрации.....</i>	69
АВТОРИЗАЦИЯ И ЗАВЕРШЕНИЕ СЕАНСА РАБОТЫ ПОЛЬЗОВАТЕЛЯ.....	70
ИЗМЕНЕНИЕ РЕГИСТРАЦИОННЫХ ДАННЫХ.....	72
ИЗМЕНЕНИЕ ПАРОЛЯ.....	72
ВОССТАНОВЛЕНИЕ ПАРОЛЯ.....	73
ОТОБРАЖЕНИЕ СПИСКА ПОЛЬЗОВАТЕЛЕЙ НА САЙТЕ.....	73
ОТОБРАЖЕНИЕ ПОЛЬЗОВАТЕЛЕЙ ПРИСУТСТВУЮЩИХ НА САЙТЕ.....	74
ЛИЧНЫЕ СООБЩЕНИЯ.....	75
<i>Нотификация.....</i>	76
АВТОРИЗАЦИЯ ПО ХЭШУ.....	76
АВТОРИЗАЦИЯ ЧЕРЕЗ ВНЕШНИЕ СЕРВИСЫ.....	79
<i>Авторизация через В Контакте.....</i>	80
<i>Авторизация через Facebook.....</i>	82
<i>Авторизация через Twitter.....</i>	85
АВТОРИЗАЦИЯ ЧЕРЕЗ USB-ТОКЕНЫ.....	87
<i>Привязка пользователя к токenu.....</i>	88
<i>Авторизация в систему администрирования через USB-токен.....</i>	89
<i>Форма авторизации по токenu на сайте.....</i>	90
ДРУЗЬЯ-ВРАГИ.....	91
<i>Описание.....</i>	91
<i>Функции модуля и константы.....</i>	93
<i>Пример использования.....</i>	94
ЛИЧНЫЙ СЧЕТ.....	95
АДМИНИСТРАТИВНАЯ ЧАСТЬ МОДУЛЯ.....	96
<i>Информация.....</i>	96
<i>Регистрация и авторизация.....</i>	96

<i>Настройки</i>	100
КЛАССЫ И ФУНКЦИИ МОДУЛЯ.....	102
<i>Класс nc_auth</i>	102
УПРАВЛЕНИЕ РЕКЛАМОЙ НА САЙТЕ.....	105
ДОБАВЛЕНИЕ БАННЕРА.....	105
СЦЕНАРИИ ПОКАЗОВ.....	105
ИСКЛЮЧЕНИЯ.....	105
ДОБАВЛЕНИЕ РЕКЛАМНОЙ ЗОНЫ.....	106
ДОБАВЛЕНИЕ РЕКЛАМНОЙ КАМПАНИИ.....	106
НАСТРОЙКА ТРАНСЛЯЦИИ БАННЕРОВ НА СТРАНИЦЕ.....	106
ГЕНЕРАЦИЯ СТАТИСТИКИ.....	107
ИНТЕРНЕТ-МАГАЗИН.....	110
ВВЕДЕНИЕ.....	110
УСТАНОВКА МОДУЛЯ.....	110
КОМПОНЕНТЫ ТОВАРОВ.....	112
ЦЕНЫ ДЛЯ РАЗНЫХ ГРУПП ПОЛЬЗОВАТЕЛЕЙ.....	112
МИНИМАЛЬНАЯ ЦЕНА.....	113
СКИДКИ.....	113
<i>Скидки на отдельные товары</i>	114
<i>Скидки на всю сумму заказа</i>	115
<i>Скидки в зависимости от суммы предыдущих покупок пользователя</i>	117
<i>Наценки</i>	117
ВАЛЮТЫ.....	117
<i>Добавление новой валюты</i>	118
НАЛОГИ.....	120
ВАРИАНТЫ ДОСТАВКИ И ОПЛАТЫ.....	120
ПОДКЛЮЧЕНИЕ ПЛАТЕЖНЫХ СИСТЕМ.....	121
<i>ASSIST</i>	121
<i>Webmoney</i>	121
<i>Яндекс-деньги</i>	122
РЕДАКТИРОВАНИЕ ЗАКАЗОВ.....	123
ДОБАВЛЕНИЕ ПОЛЕЙ В КОМПОНЕНТ «ЗАКАЗ».....	123
НАСТРОЙКА ШАБЛОНОВ ПИСЕМ.....	124
ПОПУЛЯРНЫЕ ТОВАРЫ.....	124
ОПОВЕЩЕНИЕ О ЗАКАЗЕ РАЗНЫМ МЕНЕДЖЕРАМ.....	125
ИМПОРТ ДАННЫХ ИЗ 1С.....	125
<i>Первая обработка номенклатуры 1С</i>	125
<i>Импорт из 1С8</i>	126
<i>Выгрузка информации из 1С</i>	127
ВЫГРУЗКА ЗАКАЗОВ В 1С.....	128
ВЫГРУЗКА ПРЕДЛОЖЕНИЙ В ЯНДЕКС.МАРКЕТ.....	128
ВЫГРУЗКА ПРЕДЛОЖЕНИЙ В TORGMAIL.....	128
НАСТРОЙКА КОМПОНЕНТОВ: ВЫВОД ИНФОРМАЦИИ О ТОВАРАХ.....	129
НАСТРОЙКА КОМПОНЕНТОВ: ЦЕЛОЕ И ДРОБНОЕ КОЛИЧЕСТВО ТОВАРОВ.....	130
ПРЕДУСТАНОВЛЕННЫЕ ПЕРЕМЕННЫЕ.....	130
НАСТРОЙКА ВЫВОДА КОРЗИНЫ ТОВАРОВ.....	132
ДОБАВЛЕНИЕ СПИСКА ПОПУЛЯРНЫХ ТОВАРОВ.....	132
НАСТРОЙКА ВЫВОДА КОРЗИНЫ ТОВАРОВ.....	132
НАСТРОЙКИ МОДУЛЯ.....	133
УПРАВЛЕНИЕ ССЫЛКАМИ.....	134
ОПИСАНИЕ ФУНКЦИОНАЛЬНОСТИ.....	135
НАЧАЛО РАБОТЫ.....	136
КАТАЛОГ ССЫЛОК.....	136
КОДЫ ССЫЛОК.....	137
ПРОДАЖА ССЫЛОК.....	137

Покупка ссылок.....	137
Стоп-лист.....	138
Настройка модуля.....	139
Экран «Настройки модуля».....	139
Экран «Шаблоны писем».....	140
ОБЛАКО ТЕГОВ.....	142
ТРЕБОВАНИЯ К КОМПОНЕНТАМ.....	142
Вывод облака тегов.....	143
Функции модуля.....	144
Вывод результатов при выборе тега и компонент «Теги».....	145
КАЛЕНДАРЬ.....	147
ПРАКТИЧЕСКАЯ ПОЛЬЗА И ПРИМЕНЕНИЕ.....	147
ТЕХНИЧЕСКИЕ ТРЕБОВАНИЯ.....	147
ТРЕБОВАНИЯ К КОМПОНЕНТАМ.....	147
Функции модуля.....	148
Панель управления модулем.....	149
БЛОГ И СООБЩЕСТВО.....	151
ТРЕБОВАНИЯ К КОМПОНЕНТАМ.....	151
Функции модуля.....	152
Панель управления модулем.....	154
Настройки модуля.....	155
Управление действиями в компонентах.....	155
ЗАЩИТА ФОРМ КАРТИНКОЙ.....	157
Принцип работы модуля.....	157
Настройка модуля.....	157
Защита стандартных форм.....	159
Защита формы регистрации пользователей.....	159
Защита формы добавления в модуле "Форум".....	159
Защита альтернативных форм.....	160
Функции, доступные после установки модуля.....	160
nc_captcha_formfield.....	160
nc_captcha_generate_hash.....	162
nc_captcha_verify_code.....	162
nc_captcha_image.....	163
nc_captcha_generate_code.....	163
Модификация модуля САРТСНА.....	163
Аудио каптча.....	164
МОДУЛЬ "КЭШИРОВАНИЕ".....	165
Описание модуля.....	165
Предисловие.....	165
Примеры из жизни.....	166
Принцип работы кэша.....	166
Использование функционала кэширования.....	167
Блок "nosache".....	169
Административная часть.....	171
Настройки кэша.....	171
Информация.....	174
Данные аудита.....	174
Константы модуля.....	175
ИСПОЛЬЗОВАНИЕ MEMCACHED.....	176
Требования.....	176

<i>Настройка</i>	177
МОДУЛЬ «КОММЕНТАРИИ».....	178
Описание модуля.....	178
<i>Предисловие</i>	178
<i>Примеры из жизни</i>	178
<i>Принцип работы комментариев</i>	179
<i>Использование функционала комментариев</i>	180
Функционал модуля.....	182
Административная часть.....	185
МОДУЛЬ «ЛОГИРОВАНИЕ».....	195
Описание.....	195
Административная часть.....	195
<i>Последние изменения</i>	195
<i>Константы модуля</i>	196
МОДУЛЬ «ФОРУМ».....	197
Подключение функционала форума.....	197
Разграничение прав.....	199
Функции модуля.....	199
<i>Объект списка форумов</i>	200
<i>Объект форумов</i>	203
<i>Объект топиков</i>	208
<i>Объект ответов</i>	210
<i>Структура таблиц</i>	214
Административная часть.....	217
<i>Настройки форума</i>	217
<i>Конвертер из старых версий</i>	218
Константы модуля.....	219
МИНИМАГАЗИН.....	221
Возможности и требования.....	221
Подключение модуля.....	221
<i>Массовое добавление товаров в корзину</i>	222
Заказы.....	223
Скидки.....	223
Корзина.....	224
Шаблоны отображения.....	225
<i>Элемент «положить в корзину»</i>	225
<i>Элемент «товар в корзине»</i>	226
<i>Информер непустой корзины</i>	226
<i>Информер пустой корзины</i>	226
<i>Уведомление о помещении товара в корзину</i>	226
Оповещение по почте.....	227

Модули, уже установленные в системе, необходимо настроить перед началом использования. То же касается дополнительных модулей, приобретаемых отдельно (перед настройкой их, разумеется, необходимо установить – см. главу «Установка модулей» Руководства разработчика). Настройка модулей производится в разделе «Настройки» - «Настройка модулей»: иконка в виде шестеренки открывает окно для редактирования параметров модуля.

Ниже приведены инструкции по настройке и использованию в разработке всех модулей к версии 4.5

Состав версий:

Модули	Standard	Business	Corporate	Community	E-commerce	Extra
Интерфейс разработчика	+	+	+	+	+	+
Защита форм картинкой	+	+	+	+	+	+
Поиск по сайту		+	+	+	+	+
Статистика посещений			+	+	+	+
Интернет-магазин					+	+
Управление ссылками				+		+
Форум			+	+		+
Управление рекламой на сайте						+
Личный кабинет		+	+	+	+	+
Подписки и рассылки			+	+	+	+
Голосование				+		+
Блог				+	+	+
Календарь	+	+	+	+	+	+
Облако тегов				+		+
Кэширование	+	+	+	+	+	+
Комментарии			+	+	+	+
Файл-менеджер	+	+	+	+	+	+
Логирование			+	+	+	+
Минимагазин		+	+		+	+

Голосование

Модуль позволяет проводить опросы, состоящие из одного вопроса и 10 фиксированных ответов. Имеется возможность включить альтернативный вариант ответа - текстовое поле, куда голосующий может внести свой вариант.

При создании голосования можно указать:

- право пользователя выбирать несколько вариантов ответа или только один;
- используется ли альтернативный вариант ответа;
- альтернативное значение (надпись) на кнопке голосования;
- условия голосования: один голос с одного IP, cookie, голосуют только зарегистрированные в системе пользователи.

Помимо вспомогательных скриптов, модуль содержит компонент «Опросы». После успешной установки модуля вам будет предложено создать раздел «Опросы» (ссылка «завершение установки» в разделе Модули). К этому разделу будет автоматически подключен соответствующий компонент. Вы также можете подключить этот компонент к любому другому разделу.

Управление модулем осуществляется при помощи стандартного механизма компонентов. Каждый опрос представляет собой объект определенного компонента. Соответственно, можно настраивать модуль, используя все возможные механизмы настройки компонента: изменять формат отображения результатов, выносить опрос на титульную страницу, менять внешний вид опроса в зависимости от подаваемых параметров и пр.

Используемые функции:

- **poll_alreadyAnswered(\$classID, \$PollID, \$ProtectIP, \$ProtectUsers)**
Функция проверяет, участвовал ли пользователь в указанном опросе. Проверка происходит по номеру компонента \$classID и номеру опроса \$PollID, есть ли cookie у пользователя на компьютере. Если \$ProtectIP равна 1, то проверяется и IP голосующего. Если \$ProtectUsers равна 1, то проверяется, авторизован ли пользователь (по его ID).
- **poll_percentLine (\$classID, \$PollID, \$AnswerCount, \$MaxWidth, \$template)**

Функция рисует линию, ширина которой в шаблоне \$template соответствует количеству голосов (\$AnswerCount), отданных за данный ответ. Максимальная ширина линии задается переменной \$MaxWidth. \$classID и \$PollID – номер компонента и номер опросника соответственно.

В шаблоне \$template можно использовать макросы %WIDTH% – ширина линии и %PERCENT% - количество отданных голосов за данный ответ в процентах.

Функция предназначена для использования в компоненте «Опросник». Пример использования функции:

```
".opt($f_Answer1,$f_Answer10.poll_percentLine($classID,  
$f_RowID,$f_Count1,100," (%PERCENT%)<br><img  
src=/images/line.gif width=%WIDTH  
height=10>")."<br><br>")."
```

```
".opt($f_Answer2,$f_Answer10.poll_percentLine($classID,  
$f_RowID,$f_Count2,100," (%PERCENT%)<br><img  
src=/images/line.gif width=%WIDTH  
height=10>")."<br><br>")."
```

В качестве результата мы получаем:

Да (23%)

=====

Нет (77%)

=====

- **poll_alternativeAnswer(\$classID, \$PollID)**

Функция выводит список альтернативных вариантов ответа на вопрос с номером \$PollID из компонента \$classID.

Поиск по сайту

Обзор возможностей модуля

Поиск на сайте:

- Полнотекстовый поиск с учётом синонимов, стоп-слов, с ранжированием результатов по релевантности, сортировкой по дате изменения.
- Мощный язык запросов с возможностью логических запросов, поиска фраз, поиска по шаблону и похожежности, управления весом слов в запросе.
- Полноценный (в отличие от стемминга) морфологический анализ текста на русском и английском языках.
- Выпадающие подсказки в поле поиска для быстрого перехода к результатам поиска.
- Исправление запросов, набранных в неправильной раскладке клавиатуры или с ошибками.
- Возможность полностью изменить отображение поисковых форм и результатов поиска.
- Возможность указать область поиска (поиск только в определённых разделах сайта).

Индексирование документов на сайте:

- Гибкое указание области для переиндексирования вплоть до отдельных страниц; исключение страниц по регулярному выражению.
- Периодическое переиндексирование по расписанию и по запросу.
- Гибкое управление индексируемыми областями HTML-страниц: возможность исключить из индекса части страниц.
- Извлечение данных из HTML-документов по xpath-запросу и регулярным выражением с возможностью дальнейшего поиска и сортировки результатов по этим данным.
- Присваивание различного веса словам, находящимся в разных областях документа (например, в заголовках или в тэгах META).
- Поддержка доменных имён на национальных языках (IDN) и путей с нелатинскими буквами.
- Многоязыковая поддержка: возможность подключения анализаторов и стеммеров для любых языков.
- Расширяемость: возможность подключения различных поисковых систем, сервисов проверки орфографии, поиска по файлам различных типов.

Сбор информации для администратора сайта:

- Статистика обращений к поисковой системе.
- Обнаружение ссылок на несуществующие документы (как на сайте, так и за его пределами).

Оптимизация сайта для внешних поисковых систем:

- Управление файлом robots.txt: возможность исключить разделы из результатов поиска внешних поисковых систем.
- Автоматическое создание файла sitemap.xml: задание приоритета страниц в отдельных разделах сайта, частоты изменения страниц, даты последнего изменения.

Язык запросов

По умолчанию в модуле поиска в качестве поисковой системы используется библиотека Zend_Search_Lucene

(<http://framework.zend.com/manual/en/zend.search.lucene.html>). Данная библиотека поддерживает следующие возможности в языке запросов:

- **Логические выражения в запросе** — AND (и), OR (или), NOT (не) с группировкой подзапросов:

родной AND ((город AND NOT Москва) OR деревня OR село OR посёлок)

Логические операторы должны быть набраны заглавными буквами.

Вместо AND, OR и NOT можно использовать &&, || и ! соответственно.

По умолчанию, когда между словами отсутствует логический оператор, подразумевается оператор AND. Это можно изменить в настройках модуля (вкладка «Настройки» — «Общие»).

- **Поиск документов, не включающих определённое слово:**

площадь -Красная

Запрос не может одновременно содержать логический запрос и оператор исключения «-», например, поиск по запросу площадь -Красная OR Революции приведёт к появлению сообщения об ошибке в запросе.

- **Поиск документов, включающих определённое слово** (когда в настройках модуля установлен логический оператор по умолчанию OR):
площадь +Красная Революции (найдёт все документы, имеющее слово «красный», при этом выше в результатах поиска окажутся документы со словами «площадь» и «революции», т.е. запрос эквивалентен логическому запросу Красная AND (площадь OR Революции))

— **Поиск по полю:**

title:магазин

title:(условия доставки OR обслуживания)

По умолчанию модулем поиска создаётся поля title (заголовок страницы, взятый из соответствующего тэга), meta (содержимое МЕТА-тэгов description, keywords) и last_modified (содержит время обновления документа в формате ГГГГММДДччммсс).

— **Поиск документов, содержащих фразу:**

"поиск по сайту"

Будут найдены все документы, в которых слова встречаются в указанном порядке.

— **Поиск документов, содержащих слова на определённом расстоянии друг от друга:**

"Лев Толстой"~2

Будут найдены документы, в которых слова «Лев» и «Толстой» расположены не далее, чем в двух словах друг от друга — например, документы с фразами «Лев Толстой», «Лев Николаевич Толстой», «Толстой Лев» (а также «толстые львы»).

— **Поиск похожих слов** с указанием степени схожести (от 0.1 до 0.9) или без него (по умолчанию применяется коэффициент 0.5):

тяжелый~ (найдет документы со словами «тяжелый», «тяжесть», «тяжко» и т.п.)

дом~0.9 (найдет документы, в которых есть слова, отличающиеся от слова «дом» на одну букву, например «дот», «дам», «том»)

Возможность указать, что какое-либо слово в запросе имеет бóльшую (или меньшую) важность:

купить^0.5 LCD телевизор^4

— **Поиск слов по шаблону** (шаблон для подстановки одной буквы — ?, для любого количества символов — *):

нетк?t

авто*

гео*ский

По умолчанию поиск по шаблону отключен в настройках модуля, так как может создавать повышенную нагрузку на сервер.

— **Поиск по интервалу значений:**

last_modified:[20100401000000 TO 20100402000000] (включает конечные значения интервала)

last_modified:{20100401000000 TO 20100402000000} (не включает конечные значения)

title:[картофель ТО томат] (включает слова «картофель», «лук», «огурец», «томат» и т.п.)

По умолчанию поиск по интервалу значений отключен в настройках модуля.

Запрос с интервалом переписывается в логический запрос со всеми возможными значениями, поэтому может возникнуть ошибка «слишком много терминов в запросе» (по умолчанию — 2048).

Интерфейс модуля в панели управления сайтом

В системе администрирования модуль содержит 4 вкладки:

- Общие
- Индексирование
- Списки
- Настройки

Общие

Данная страница показывает общую статистику модуля поиска.

Индексирование

Данная страница показывает все правила индексации с возможностью запустить переиндексацию в отдельном окне или в фоновом режиме.

Поиск по сайту

Информация **Индексирование** Списки Настройки

Сайт 1 [\[подробнее\]](#)
Правило ни разу не выполнялось.
[Индексировать в реальном времени](#) [Индексировать в фоновом режиме](#) [Редактировать](#)

Новости [\[подробнее\]](#)
Правило ни разу не выполнялось.
[Индексировать в реальном времени](#) [Индексировать в фоновом режиме](#) [Редактировать](#)

Списки

Данная страница содержит 5 подвкладок:

- запросы
- битые ссылки
- синонимы
- стоп-слова
- события

Запросы

Показывает все поисковые запросы с указанием времени последнего запроса, количеством запросов по определенной фразе и о количестве найденных страниц. Также есть фильтр по фрагменту запроса, временного промежутка и результату.

Битые ссылки

Страница показывает все битые ссылки, обнаруженные при индексации.

Синонимы

Страница показывает все синонимы. Также здесь можно добавить новые синонимы.


Поиск по сайту

Информация

Индексирование

Списки

Настройки

Запросы

Битые ссылки

Синонимы

Стоп-слова

События

Любой язык ▾

Язык	Синонимы	Изменить	
ru	NETCAT НЕТКЭТА		

Стоп-слова

Страница показывает все стоп-слова, а так же позволяет производить действия над ними: добавление, изменение, удаление.

События

Выводит лог работы модуля. В журнал, в зависимости от настроек, можно записывать:

- Ошибка в работе модуля
- Исключение
- Ошибка РНР
- Предупреждение РНР
- Запуск планировщика
- Начало, окончание индексирования
- HTTP-запрос
- Невозможность определения раздела документа
- Добавление ссылки в очередь
- Краткие сведения о полученном документе
- Полные сведения о полученном документе

Области индексирования

Задание списка страниц, подлежащих индексированию

Модуль поиска позволяет гибко указывать область для переиндексирования (вкладка «Настройки» — «Правила» в панели управления модулем).

1. Указание сайта:

site1 — сайт с идентификатором 1

mysite.ru — домен сайта (**не указывайте** протокол http!)

allsites — все сайты под управлением данной копии NetCat

2. Указание раздела:

sub10* — раздел с идентификатором 10 и все его потомки (разделы, объекты)

sub10 — раздел с идентификатором 10 и страницы объектов, расположенные в нём

sub10. — раздел с идентификатором 10, только главная страница раздела

Вместо идентификатора раздела в формате subXX можно указать путь к разделу, заканчивающийся на «/» (также можно использовать модификаторы «.», «*»):

/company/news/

/company/news/*

/company/news/.

3. Указание конкретной страницы — необходимо указать путь до страницы:

/company/about/history.html

Для исключения подобласти необходимо поставить знак «-» перед её описанием:

-sub10

-/company/*

Можно указать любое количество подобластей через пробел, перенос строки или запятую:

/company/* /catalogue/* -sub15

Если необходимо исключить один или несколько уровней иерархии, разделы должны быть перечислены в порядке увеличения глубины их положения в дереве разделов:

/company/* -/company/staff/* /company/staff/director.html

В панели управления модулем («Настройки» — «Общие») можно указать глобальные (действующие для всех правил) шаблоны путей, по которым страницы будут исключены из процесса индексирования. Данные шаблоны являются регулярными выражениями и применяются к полному пути страницы. Правила задаются без разделителей и модификаторов регулярных выражений. Примеры правил:

[?&]template=59 — исключить все пути, содержащие ?template=59 или &template=59

^http://[^/]+/netcat/ — исключить все пути на всех сайтах, начинающиеся с /netcat/

\.rss\$ — исключить все пути, заканчивающиеся на .rss
Не забудьте экранировать спецсимволы регулярных выражений «.», «?». Прямой слеш «/» экранировать не нужно.

Все страницы, пути которых попадают под правила исключения, не будут запрашиваться модулем в процессе индексирования. Правила исключения также применяются к путям на других сайтах, если включена опция проверки ссылок за пределами сайта.

Кроме того, на область поиска влияет настройка сайтов и разделов «Запретить индексирование». Все разделы, индексирование которых запрещено, не будут запрашиваться модулем в процессе переиндексирования.

Если на HTML-странице имеется тэг `<meta name="robots" content="noindex" />` (такой тэг добавляется автоматически для страниц, находящихся в запрещённых к индексированию разделах), страница также не будет индексироваться.

Исключение страницы из области индексирования также означает, что с данной страницы не будут получены для проверки ссылки на другие страницы.

Указание областей HTML-страниц, подлежащих индексированию

В панели управления модулем на вкладке «Настройки» — «HTML-документы» можно указать, какие части HTML-страниц будут индексироваться, а значит, будут доступны для поиска.

По умолчанию индексируется содержимое HTML-страниц между комментариями `<!-- content -->` и `<!-- /content -->`, а если их нет — всё содержимое тэга `body`.

Из содержимого будут исключены все части страниц, указанные в поле «Области, содержимое которых не индексируется». По умолчанию это содержимое между комментариями `<!-- noindex -->` и `<!-- /noindex -->`, содержимое тэгов `script` и `style`.

Вы можете указать собственные области, подлежащие индексированию или исключаемые из него:

По имени тэга, например: `aside`

По xpath-запросу, например: `//div[@class="menu"]`

(Обратите внимание: значения атрибутов являются регистрозависимыми, поэтому запрос, указанный выше в примере, не будет выбирать содержимое элемента `<div class="Menu">`.)

По регулярному выражению, например: `#<!--junk-->(.*?)<!--/junk-->#i`

(Обратите внимание: разделителем регулярного выражения должен быть символ «#».)

В процессе обработки HTML-страниц они преобразовываются в объект класса `DOMDocument` (см. <http://php.net/domdocument>), поэтому их исходный код в момент выполнения xpath-запросов и регулярных выражений может отличаться от полученного с сайта:

- будут закрыты все тэги;
- будут удалены лишние закрывающиеся тэги;
- все кавычки будут заменены на двойные;
- символ «&» будет заменен на «&»;
- тэги без содержимого не будут иметь закрывающего слеша, например `<br />` будет преобразован в `<br>`;
- переносы строк и отступы могут быть изменены.

Например, некорректный с точки зрения стандартов фрагмент `<div><b>text</div></b>` будет преобразован в `<div><b>text</b></div>`.

В связи с указанными особенностями могут возникать ошибки при индексировании страниц с тэгом `<script>` с кодом, выводящим тэги `</body>` и `</html>`. В случае, когда подобный код необходим, можно разбивать данные тэги: `document.write("</" + "body">")`.

Сохранение частей страниц для использования в результатах поиска

При использовании стандартной службы поиска имеется возможность сохранять фрагменты HTML-страниц в отдельные поля (вкладка «Настройки» — «HTML-документы», блок «Извлечение данных»). Эти фрагменты могут быть получены на странице с результатами поиска из массива `$f_meta[имя_поля]`, использованы для сортировки результатов (например, сортировка товаров по цене или другой характеристике) или для организации простого фасетного поиска (например, поиск по определённому автору). Для выбора фрагментов можно использовать имена

тэгов, xpath-запросы и регулярные выражения, аналогично указанию фрагментов, подлежащих индексированию (см. выше).

Каждое поле данных имеет настройки:

- тип поля: строка или целое число (от 0 до 999.999.999);
- если поле *участвует в поиске*, его значение сохраняется в поисковом индексе, и при выполнении поискового запроса будет производиться поиск по данному полю (в том числе, когда перед поисковой фразой имя поля не указано);
- если установлен параметр *«значение доступно в результатах поиска»*, изначальное значение соответствующего фрагмента HTML-страницы можно получить в результатах поиска;
- если установлен параметр *«нормализовывать текст»*, будет доступен поиск с учётом морфологии;
- параметр *разрешить сортировку по полю* также включает возможность поиска по полю.

Пример: поиск по каталогу товаров (если при этом полнотекстовый поиск не нужен, следует использовать стандартные механизмы фильтрации результатов NetCat — см. раздел «Поиск и выборка» в «Руководстве разработчика») с возможностью фильтрации и сортировки по цене.

На странице с выводом **полной информации** о товаре необходимо добавить разметку для извлечения необходимой информации (в данном случае — цены). Также добавим код, подсказывающий модулю поиска, что это страница с товаром:

```
<div data-type="goods">
```

Описание...

Цена: >20000 руб.

```
</div>
```

Добавим информацию о полях на вкладке «Настройки» — «HTML-документы»:

- 1) Имя поля: `page_type`, запрос: `//div[@data-type]/@data-type` (получить значение атрибута `data-type` тэга `div`), тип: строка, поле участвует в поиске, недоступно в результатах, не нормализуется, не участвует в сортировке.
- 2) Имя поля: `price`, запрос: `//span[@class='price']` (получить содержимое тэга `span` с классом `price`), тип: целое число, поле участвует в поиске, доступно в результатах, не нормализуется, участвует в сортировке.

После переиндексации страниц с товарами появится возможность:

- выводить цену товара в результатах поиска. В компоненте поиска в поле «Объект в списке» цена будет доступна в массиве `$f_meta['price']`.
- ограничить поиск только товарами (если включена возможность поиска по полю, вкладка «Настройки» — «Общие»). Для этого необходимо модифицировать запрос (например, в системных настройках компонента поиска), дописав к нему `page_type:goods`.
- сортировать результаты по цене (должен быть включен поиск по полю), передав GET-параметр `sortby=price` при обращении к компоненту поиска (`sortdirection=asc` — сортировка по возрастанию, `sortdirection=desc` — по убыванию; по умолчанию результаты сортируются по убыванию).
- искать товары по диапазону цен (должен быть включен поиск по интервалу значений), например, для поиска товаров с ценой от 1000 до 5000 нужно дописать к поисковому запросу условие `page_type:goods price:[1000 TO 5000]`.

Большое число дополнительных полей, по которым разрешен поиск, может замедлить поиск, поэтому не стоит злоупотреблять этой возможностью.

Области поиска на сайте

При задании области поиска на сайте применяются те же правила, что и при задании областей индексирования, *за исключением* возможности указать отдельные страницы (например, «`/company/.`» или «`/company/about/history.html`»). Кроме того, можно использовать две специальные области:

- `thissub` — поиск в текущем разделе;
- `parentsub` — поиск в родительском разделе.

К `thissub` и `parentsub` также можно добавлять модификатор «*», например область `parentsub*` для формы поиска, размещённой в разделе `/events/exhibitions/`, идентично указанию области `/events/*`.

Запуск по расписанию. Постановка заданий в очередь

Для того, чтобы на сайте работал поиск, необходимо вначале проиндексировать этот сайт. Изменения не будут учитываться в результатах поиска до тех пор, пока не будет произведено переиндексирование изменившихся страниц.

Индексирование может выполняться в двух режимах: в фоновом (запуск по расписанию или требованию через планировщик операционной системы или инструмент «Управление задачами» NetCat) или по запросу из панели управления модулем (в окне браузера).

Для индексирования в фоновом режиме необходимо настроить запуск индексатора в *cron* (**предпочтительный способ**) или в «Управлении задачами».

Настройка для запуска индексирования по расписанию через *cron*

В UNIX-совместимых операционных системах запуск задач производится через утилиту *cron*. Как правило, добавить задание для *cron* можно в панели управления хостингом (или, если этой возможности нет, через SSH-доступ). Необходимо добавить запуск планировщика индексирования модуля поиска, например, каждую минуту (т.е. каждую минуту будет производиться поиск задач по индексированию сайтов):

```
php /путь/к/корню/сайта/netcat/modules/search/indexing/console.php
```

Настройка для запуска индексирования по расписанию через «Управление задачами»

Если по какой-либо причине вы не можете добавить запуск планировщика индексирования напрямую в *cron*, добавьте задачу на странице «Управление задачами» (меню «Инструменты») со следующей ссылкой на скрипт:

```
/netcat/modules/search/indexing/netcat_cron.php?secret_key=XXXX
```

Вместо XXXX необходимо подставить значение, указанное на вкладке «Настройки» — «Системные» в поле «IndexerSecretKey».

Убедитесь, что в расписание *cron* на вашем сервере добавлен вызов скрипта `netcat/admin/crontab.php`, иначе индексирование не будет производиться.

Правила индексирования

На вкладке «Настройки» — «Правила» в панели управления модулем можно задать правила индексирования. Каждое правило описывает индексируемую область и частоту её переиндексирования.

О формате задания областей переиндексирования см. выше в разделе «Задание списка страниц, подлежащих индексированию».

Период повторения задачи переиндексирования может быть задан в минутах, часах, днях, или по дню месяца. При этом параметр «начиная с»

для периодов, указанных в минутах и часах, означает, что правило будет в первый раз выполнено в указанное время (в день создания или изменения правила или на следующий день, если время уже прошло), а затем круглосуточно через указанный интервал.

Постановка задачи переиндексирования в очередь

Любое правило можно запланировать к исполнению в указанное время; при этом, если речь идёт о периодически исполняемом правиле, расписание его выполнения остаётся без изменения.

По умолчанию если в очереди имеется несколько заданий на переиндексирование одной и той же области с интервалом менее 5 минут, индексирование будет выполнено однократно (параметр «MinScheduleInterval» на вкладке «Настройки» — «Системные», значение задаётся в секундах).

Для постановки в очередь используется метод:

```
nc_search::index_area(string $area_string = "allsites", string $when = "now")
```

Параметр `$area_string` может содержать ID правила индексирования или указание области (см. выше раздел «Области поиска на сайте»).

Параметр `$when` может принимать любые значения, допустимые для функции PHP `strtotime()`.

Примеры:

```
nc_search::index_area(1); // выполнить правило с ID=1 при первой возможности
```

```
nc_search::index_area("/news/*", "tomorrow"); // переиндексировать /news/* в полночь
```

Пример использования в поле «Действие после изменения» компонента:

```
// переиндексировать раздел, в котором находится объект, и его подразделы:
```

```
nc_search::index_area("sub$sub*");
```

```
// переиндексировать только страницу с объектом:
```

```
nc_search::index_area(nc_message_link($message, $classID));
```

Любой процесс переиндексирования также запускает проверку ссылок, находящихся на индексируемых страницах, в соответствии с настройками модуля.

Интеграция модуля в макеты дизайна сайта

Для интеграции модуля с макетами дизайна сайта существует вспомогательный класс `nc_search_ui`. Копия экземпляра класса находится в глобальной области видимости переменных под именем `$nc_search`. Чтобы воспользоваться ею в макете дизайна, нужно добавить в системные настройки макета строку:

```
global $nc_search;
```

Также вы можете получить ссылку на этот объект при помощи метода `nc_search::get_object()`:

```
$nc_search = nc_search::get_object();
```

Простая форма поиска

Простая форма поиска может содержать следующие элементы: поле для ввода поискового запроса, кнопку отправки запроса, элементы для выбора области поиска, выпадающее поле с подсказками поисковых запросов. Форму поиска можно вывести в любой части сайта (обычно — в макете дизайна), используя метод `show_form()`:

```
$nc_search->show_form([string $params[, array $template[, array $areas]]])
```

Параметры:

- `$params` — строка с дополнительными параметрами формы, соединёнными через амперсанд (например, `"inputsize=30&suggest=0"`). Если значения содержат символы `&`, `=`, пробел, они должны быть указаны как `%26`, `%3D`, + соответственно. Возможные параметры:
 - `submitname` — текст кнопки отправки запроса.
 - `inputsize` — размер поля ввода. По умолчанию — 20.
 - `suggest` — если значение равно 1 или `on`, будет использоваться автозаполнение (должно быть разрешено в настройках модуля). По умолчанию — 1.
 - `areatype` — индекс элемента массива с шаблоном формы, отвечающего за вывод списка областей поиска:
 - `empty` — не выводить области поиска.
 - `select` — вывести список областей поиска в виде выпадающего списка.
 - `radio` — вывести список областей поиска в виде переключателей (`radiobutton`).

- `checkbox` — вывести список областей с «галочками» (`checkbox`). Удобно для использования вместе с областью поиска `thissub` (поиск в только текущем разделе).
- `links` — вывести области поиска в виде ссылок. Нажатие на ссылку с областью поиска перезагружает страницу поиска с поисковым запросом, заданным при загрузке страницы. Удобно для использования на странице поиска.

Значение по умолчанию: если параметр `$areas` задан — `select`, иначе — `empty`.

- `showadvancedlink` — выводить ли ссылку на форму расширенного поиска (если её использование разрешено в настройках модуля). По умолчанию — 0 (не выводить ссылку).
- `inputvalue` — значение в поле ввода, по умолчанию равен GET-параметру `query`.
- `selectedarea` — выбранная область; может быть перечислено несколько областей через запятую. По умолчанию равен GET-параметру `area`.
- `actionurl` — путь до раздела поиска; по умолчанию определяется автоматически.

— `$template` — массив с альтернативным шаблоном формы поиска. Может содержать элементы `prefix` (начало формы — код до вывода областей поиска), `suffix` (окончание формы — код после вывода областей поиска), элементы для вывода областей поиска: `unactive` (невыбранная область поиска), `active` (выбранная область поиска), `divider` (разделитель между областями поиска). В массиве `$template` могут быть заданы не все элементы, а только их часть; недостающие элементы будут взяты из описания формы поиска по умолчанию в настройках модуля.

В шаблонах вывода областей поиска можно использовать псевдопеременные `%AREANUM` (порядковый номер области поиска), `%AREA` (область поиска), `%NAME` (текстовое описание области поиска), `%URL` (ссылка на страницу поиска с поисковым запросом, но без параметра `area`). Во всех элементах шаблона можно использовать параметры-переменные из первого аргумента метода, например, `$inputsize` или `$areastype`.

Структура формы поиска по умолчанию определяется в панели управления модулем на вкладке «Настройки» — «Отображение».

Как правило, изменение шаблона не требуется, поскольку внешний вид формы можно изменить при помощи CSS.

- `$areas` — массив с областями поиска, где ключ — описание области (см. выше в разделе «*Области поиска на сайте*»), а значение — текстовое описание, которое будет выведено в форме. Примеры:

```
array("thissub" => "только в этом разделе")
array("" => "на всём сайте",
      "/catalogue/*" => "в каталоге товаров",
      "sub30*" => "в новостях")
```

Расширенная форма поиска

Настройки расширенной формы поиска задаются на вкладке «Настройки» — «Отображение» (разделы «Расширенный поиск» и «Шаблон отображения расширенной формы поиска»).

Расширенная форма поиска выводится при помощи метода:

```
$nc_search->show_advanced_form([array $template])
```

Параметр метода — массив с шаблоном отображения формы, который может содержать следующие элементы:

- `prefix` — начало формы.
- `input` — основное поле ввода.
- `exclude` — строка с полем для ввода слов, которые нужно исключить из результатов поиска.
- `field` — строка с выбором поля, в котором производится поиск.
- `interval` — строка с выбором интервала последнего изменения документа.
- `suffix` — окончание формы.

Можно указать не все элементы, при этом отсутствующие элементы будут взяты из шаблона, указанного в панели управления модулем.

Вывод результатов поиска

Вывод результатов поиска осуществляется при помощи служебного компонента «Поиск по сайту», расположенного в одноимённой группе компонентов.

Для получения результатов в системных настройках компонента используется метод

```
$nc_search->get_results(string $query_string[, string|array $area[, string $params]])
```

Описание параметров:

- `$query_string` — поисковый запрос
- `$area` — область поиска в виде строки или массива с областями поиска
- `$params` — дополнительные параметры в виде строки через амперсанд:
 - `field` — название поля, в котором производится поиск. Игнорируется, если в настройках модуля отключен поиск по полю.
 - `interval` — должен содержать непустое значение для срабатывания фильтрации результатов по дате последнего изменения.
 - `intervalunit` — тип интервала, по которому производится фильтрация результатов. Допустимые значения: `hour`, `day`, `week`, `month`.
 - `intervalvalue` — значение интервала (число).
 - `sortby` — поле, по которому производится сортировка. Если пустое значение — результаты сортируются по релевантности. Игнорируется, если в настройках модуля отключен поиск по полю. Допустимые значения: `lastupdated`.
 - `sortdirection` — направление сортировки. Допустимые значения: `desc` (сортировка по убыванию), `asc` (сортировка по возрастанию). Значение по умолчанию — `desc`.
 - `language` — язык запроса. По умолчанию определяется автоматически (по языку раздела, в котором расположен компонент поиска).
 - `curPos` — номер первого результата (для перехода по страницам результатов).
 - `recNum` — количество результатов на странице, по умолчанию 10 (берется из настроек компонента в разделе).
 - `correct` — если значение равно 1, модуль поиска будет пробовать исправлять запросы, не давшие результатов (по умолчанию равно значению соответствующей настройки модуля). Игнорируется, если в настройках модуля отключено исправление запросов.

Оформление результатов поиска можно изменить в полях раздела «Шаблон отображения списка объектов», однако, как правило, этого не требуется: оформлением можно управлять при помощи CSS-стилей.

Результаты выполнения поискового запроса сохраняются в объекте класса `nc_search_result` (переменная `$results`), у которого доступны следующие методы:

- `$results->get_error_message()` — получить сообщения об ошибках, если они возникли при выполнении запроса.
- `$results->get_correction_suggestion()` — получить сообщение об исправлении запроса, если таковое было произведено.
- `$results->get_search_time()` — получить время в секундах, затраченное на выполнение запроса.

В поле «Объект в списке» можно использовать следующие свойства документа:

- `$f_url` — адрес страницы
- `$f_title` — заголовок страницы
- `$f_context` — фрагменты текста, совпавшие с поисковым запросом. Поле недоступно, если в настройках модуля (вкладка «Настройки» — «Отображение») отключена опция «показывать фрагмент текста в списке найденных страниц». Длина фрагмента определяется настройкой «Максимальное количество слов в фрагменте текста» на вышеупомянутой вкладке.
- `$f_lastmodified` — время последнего изменения (в виде строки в формате «ГГГГ-ММ-ДД ЧЧ:ММ:СС»)
- `$f_content` — проиндексированный текст документа
- `$f_language` — код языка документа
- `$f_contenttype` — MIME-тип документа (например, "text/html")
- `$f_site_id` — идентификатор сайта, на котором расположен документ
- `$f_meta` — массив с дополнительными полями (см. раздел «Сохранение частей страниц для использования в результатах поиска»)

Стилизация списка подсказок

Выпадающий список подсказок (с заголовками страниц или запросами, подходящими под введенный пользователем запрос) выводится при помощи библиотеки jQuery UI (компонент «Autocomplete»), поэтому для работы такого списка на странице должен быть подключен скрипт jQuery.

Об описании процесса стилизации списка подсказок можно прочитать на странице документации jQuery UI: <http://jqueryui.com/demos/autocomplete/#theming>.

Базовый набор CSS-стилей, необходимый для корректной работы выпадающего списка, имеется в файле `/netcat/modules/search/suggest/autocomplete.css`. Рекомендуется включить содержимое этого файла в набор стилей вашего сайта.

Структура выпадающего списка следующая:

```
<ul class="ui-autocomplete">
  <li class="ui-menu-item"><a>Элемент 1</a></li>
  <li class="ui-menu-item"><a>Элемент 2</a></li>
  ...
</ul>
```

Разработка расширений модуля

Существует возможность изменять поведение модуля поиска на различных этапах индексирования и поиска путём подключения расширений.

Обзор архитектуры модуля

Процесс поиска состоит из двух этапов — сохранения в поисковом индексе (индексирования) и извлечения документов из него (собственно поиска).

I. Индексирование:

- 1) Скачивание документов с сайта.
- 2) Обработка полученных документов: извлечение данных, ссылок на другие документы:
 - чтение содержимого документа в зависимости от его формата;
 - принятие решения о том, следует ли индексировать данный документ (в зависимости от содержимого);
 - получение содержимого, подлежащего индексированию, извлечение метаданных;
 - разбивка содержимого на слова, фильтрация стоп-слов, приведение слов к базовой форме для поиска с учётом морфологии языка.
- 3) Сохранение полученных данных в поисковом индексе.

II. Поиск: извлечение из поискового индекса документов, соответствующих запросу пользователя.

- 1) Приведение слов в запросе к базовой форме.
- 2) Выполнение запроса, сортировка результатов.
- 3) Попытка исправления запроса, если результаты не получены.

Модуль поиска состоит из следующих частей:

1. **Поставщик службы поиска** (интерфейс `nc_search_provider`).
Обеспечивает хранение документов, полнотекстовый поиск по ним;

также может управлять способом загрузки документов (служба поиска может реализовать получение документов самостоятельно) и планированием заданий переиндексирования. По умолчанию используется библиотека `Zend_Search_Lucene`; дополнительные свойства документа (такие как полный текст документа, его тип и т.п.) хранятся в СУБД в таблице `Search_Document` для увеличения быстродействия и удобства организации процесса переиндексирования.

2. **Планировщик переиндексирования** (`nc_search_scheduler`). Отвечает за запуск переиндексирования областей согласно расписанию.
3. **Индексатор** (по умолчанию — `nc_search_indexer`). Отвечает за обработку задачи переиндексирования: обеспечивает загрузку всех страниц, находящихся в очереди текущей задачи переиндексирования, обработку полученных данных и их сохранение в поисковом индексе.
4. **Обработчики документов** (интерфейс `nc_search_document_parser`). Обеспечивают извлечение данных из документов различных типов (например, HTML-документов, текстовых файлов и т.п.).
5. **Анализаторы текста** (наследуют абстрактный класс `nc_search_language_analyzer`). Обеспечивают приведение слов к базовой форме (морфологический анализ или стемминг).
6. **Текстовые фильтры** (наследуют абстрактный класс `nc_search_language_filter`). Отвечают за приведение слов во всех документах и поисковых запросах к нормальной (базовой) форме.
7. **Корректировщики пользовательских запросов** (наследуют абстрактный класс `nc_search_language_corrector`): орфографическая проверка введенного запроса или иные способы исправления запросов, не давших результатов.

Все части модуля, реализующие интерфейсы или наследуемые от указанных в списке абстрактных классов, могут быть заменены на пользовательские. Все эти интерфейсы и абстрактные классы расширяют интерфейс `nc_search_extension`.

За подключение расширений модуля отвечает класс `nc_search_extension_manager`. Когда модулю необходимо получить объект, реализующий какую-либо функциональность, происходит вызов метода `nc_search_extension_manager::get($extension_interface, $context)`. Объект `$context` содержит дополнительные сведения о контексте операции, такие как MIME-тип документа и его язык. Менеджер расширений подбирает список подходящих под контекст расширений и возвращает их в виде объекта `nc_search_extension_chain`. Расширения могут быть применены к исходным данным последовательно при помощи метода `apply()` (например,

в случае текстовых фильтров); может быть взято расширение с высшим приоритетом (например, будет использован только один — первый — обработчик для каждого из типов документов).

Для добавления расширения необходимо написать класс, реализующий соответствующий интерфейс, и добавить его в панели управления модулем в список на вкладке «Настройки» — «Расширения», указав условия его использования (язык документа, тип документа, активная служба поиска).

Обработчики документов различных типов

Имеется возможность индексировать файлы различных типов. Обработчик документа должен реализовать интерфейс **nc_search_document_parser**:

— public function **load**(nc_search_indexer_crawler_response \$response): void

После загрузки документа индексатором будет вызван метод `$parser->load()`. Обработчик создается во время сессии индексирования однократно для каждого из типов документа, переключение на следующий документ производится вызовом метода `load()`, поэтому не забудьте сбросить в этом методе относящиеся к предыдущему документу свойства, если таковые имеются.

В качестве параметра передается экземпляр класса

`nc_search_indexer_crawler_response`, у которого доступны следующие методы:

- `$response->get_body()` — получение тела ответа HTTP (содержимого документа)
- `$response->get_header('название-заголовка-строчными-буквами')` — получение заголовка ответа HTTP
- `$response->get_url()` — получение адреса страницы
- `$response->get_content_type()` — получение типа документа

От обработчика зависит, что он будет делать с полученной информацией (например, может быть целесообразным использовать не содержимое ответа, а считать файл с локальной файловой системы).

— public function **should_index**(): boolean

Обработчик должен вернуть TRUE, если загруженный документ подлежит индексированию. (Обработчик может принять решение о том, что документ не следует индексировать, например, исходя из каких-либо метаданных, имеющихся в документе.)

— public function **get_document**(): nc_search_document

Метод должен вернуть экземпляр класса `nc_search_document` (далее — «документ»).

Каждый документ состоит из полей. Служба поиска сохраняет их в поисковом индексе и (или) в СУБД. У документа как минимум должны быть установлены значения полей `title` (заголовок документа) и `content` (содержимое):

```
$doc = new nc_search_document();
$doc->set_field_value('title', 'Заголовок');
$doc->set_field_value('content', 'Текст документа');
```

Документ может содержать и другие поля, которые будут использоваться при поиске — например, поля с бóльшим или меньшим весом:

```
$field = new nc_search_field(array(
    'name' => 'important_field',    // имя поля
    'weight' => 3,                  // вес поля
));
$doc->add_field($field);
$doc->set_field_value('important_field', 'текст с весом
3');
```

— `public function extract_links(): array`

Метод должен вернуть массив с гиперссылками, найденными в документе или **пустой массив**, если ссылок не найдено.

Образец скелета класса для индексирования PDF-документов:

```
<?php
// укажите путь до папки с модулем поиска (в данном примере
предполагается,
// что класс находится в файле
modules/default/function.inc.php):
require_once dirname(__FILE__) . '/../search/function.inc.php';

class my_pdf_parser implements nc_search_document_parser {
    protected $content;
    protected $file_name;

    public function load(nc_search_indexer_crawler_response
$response) {
        $this->file_name = basename($response->get_url());
        // тут нужно каким-то образом получить текстовое содержимое
документа:
        $this->content = pdf_get_content($response->get_body());
    }

    public function should_index() {
```

```

    return true;
}

public function extract_links() {
    return array();
}

public function get_document() {
    $doc = new nc_search_document();
    $doc->set_field_value('title', $this->file_name);
    $doc->set_field_value('content', $this->content);
    return $doc;
}
}

```

После создания класса нужно добавить запись о нём на вкладке «Настройки» — «Расширения». Для примера выше (обработка PDF-файлов) значения будут следующими: интерфейс расширения — `nc_search_document_parser`, класс расширения — `my_pdf_parser`, MIME-тип — `application/pdf`. После этого все документы, которые ваш веб-сервер отдаёт с заголовком `Content-type: application/pdf` будут обрабатываться указанным расширением.

Анализаторы текста

Анализаторы текста (наследуют `nc_search_language_analyzer`) отвечают за приведение слов к базовой форме.

По умолчанию для морфологического анализа русского и английского языков используется библиотека `phpMorphy` (<http://phpmorphy.sf.net/>).

Библиотека также имеет словари для украинского, эстонского, немецкого языков (не поставляются с NetCat). Для этих языков необходимо скачать соответствующие UTF-8 словари с сайта `phpMorphy`, поместить их в папку `modules/search/lib/3rdparty/phpmorphy/dicts/`, и добавить запись на вкладке «Настройки» — «Расширения» (интерфейс расширения — `nc_search_language_analyzer`, класс расширения — `nc_search_language_analyzer_morphy`, язык — `Ukrainian`, `Estonian` или `German`).

Для создания стеммера можно создать класс, наследующий `nc_search_language_stemmer`, в этом случае необходимо реализовать только метод `stem(string $word)`.

Обратите внимание, что на вход анализатору (при настройках по умолчанию) попадают слова в верхнем регистре, в том же регистре должен быть результат выполнения обработки текста анализатором.

Текстовые фильтры

Текстовые фильтры (наследуют `nc_search_language_filter`) обрабатывают текст как на этапе индексирования, так и при обработке поисковых запросов. Результатом применения цепочки фильтров является набор слов в том виде, в котором они подлежат индексированию (отсечены короткие и стоп-слова, слова приведены к одному регистру, находятся в базовой форме).

Например, если перед вами стоит задача организовать поиск текста на французском языке без учёта диакритических знаков (диакритические знаки не обязательны для заглавных букв, поэтому слова в верхнем регистре могут не находиться при поиске), вы можете создать текстовый фильтр:

```
<?php
class my_diacritics_filter extends nc_search_language_filter {
    public function filter(array $words) {
        foreach ($words as $i => $word) {
            $words[$i] = strtolower($word, "ÀÂÇÈÉÊËÎÏÔÙÛÜÝ",
"AACEEEEEIIIOUUUY");
        }
        return $words;
    }
}
```

Созданный фильтр необходимо добавить в список на вкладке «Настройки» — «Расширения», указав соответствующий язык и подобрав приоритет таким образом, чтобы фильтр находился между фильтрами `nc_search_language_filter_case` и `nc_search_language_filter_analyzer` (приоритет от 61 до 99).

Корректировщики запросов

Корректировщики запросов используются для исправления запросов, поиск по которым не дал результатов (например, если запрос набран в неправильной кодировке или с орфографическими ошибками).

Корректировщик запросов должен быть наследником класса `nc_search_language_corrector` и реализовывать метод `correct()`:

— `public function correct(nc_search_language_corrector_phrase $phrase): boolean`

Возвращает `TRUE`, если запрос был исправлен, иначе — `FALSE`.

Параметр — экземпляр класса `nc_search_language_corrector_phrase`, позволяющий получить как запрос целиком (метод `to_string()`), так и составляющие его слова (метод `get_terms()`).

Все имеющиеся корректировщики вызываются по порядку (в соответствии с приоритетом) до тех пор, пока исправленный запрос не даст результатов, поэтому часть слов в запросе уже может быть исправлена. Чтобы получить только неисправленные (некорректные) слова, используйте метод `get_not_corrected_terms()`.

Методы `get_terms()` и `get_not_corrected_terms()` возвращают массив с экземплярами объектов `nc_search_language_corrector_phrase_term`. Для получения слова в том виде, как оно было введено пользователем, следует использовать метод `$term->get('term')`, для сохранения исправленного варианта — `$term->set_corrected($corrected_word)`.

Минималистический пример использования библиотеки проверки орфографии *pspell* (<http://php.net/pspell>):

```
<?php
class my_pspell_corrector extends nc_search_language_corrector
{
    public function correct(nc_search_language_corrector_phrase
$phrase) {
        $pspell_link = pspell_new($this->context->get('language'));
        $has_corrections = false;
        foreach ($phrase->get_not_corrected_terms() as $term) {
            $suggestions = pspell_suggest($term->get('term'));
            if ($suggestions) {
                $term->set_corrected($suggestions[0]);
                $has_corrections = true;
            }
        }
        return $has_corrections;
    }
}
```

Подключение других поисковых систем

Существует возможность заменить библиотеку *Zend_Search_Lucene* на другой поисковый механизм, например, *Sphinx*, *Solr* или даже внешнюю поисковую систему, такую как *Яндекс.XML* или *Google Site Search*. Такая потребность может возникнуть в случае неудовлетворительной производительности *Zend_Search_Lucene* на медленном сервере, при большой нагрузке или значительном объеме проиндексированной информации.

Для этого необходимо создать класс-адаптер для поисковой системы, реализующий интерфейс `nc_search_provider`. При этом, в зависимости от возможностей поискового сервера, можно использовать или не использовать возможности, заложенные в модуль поиска (индексатор и

HTTP-робот, анализаторы и фильтры текста, корректировщики запросов и т.п.).

Методы интерфейса **nc_search_provider**:

— public function **first_run**(): void

Метод вызывается однократно только при первом запуске модуля после его установки; в нём можно выполнить действия, необходимые для начала работы с поисковой системой. Метод не вызывается при смене поисковой системы в настройках модуля, поэтому для написанного вами адаптера достаточно создать пустой метод `first_run()`.

— public function **process_document**(nc_search_document \$document): void

Обработать документ (конкретный класс должен проверить, есть ли документ в индексе, и в зависимости от этого обновить существующий документ или добавить новый).

— public function **remove_document**(nc_search_document \$document): void

Удаление документа из индекса.

— public function **commit**(): void

Запись проделанных изменений в индекс (используется, когда при обработке документа в методах `process_document()` и `remove_document()` запись изменений на диск не производится, изменения сохраняются в памяти).

— public function **optimize**(): void

Запуск оптимизации индекса, если таковая необходима (выполняется после окончания переиндексирования области).

— public function **find**(nc_search_query \$query[, boolean \$should_highlight = true]): nc_search_result

Выполнение запроса `$query`. Параметр `$should_highlight` указывает, следует ли подсвечивать совпавшие слова в результатах. Метод должен вернуть объект класса `nc_search_result`, содержащий информацию о найденных документах в виде объектов класса `nc_search_result_document`. Для получения текста поискового запроса используйте метод `$query->to_string()`.

— public function **index_area**(string \$area_string[, int \$run_type = nc_search::INDEXING_NC_CRON]): void

Запуск переиндексирования области.

- `$area_string` — идентификатор правила или описание индексируемой области
- `$run_type` — тип запуска: запуск из консоли (`nc_search::INDEXING_CONSOLE`), «Управления задачами» NetCat (`nc_search::INDEXING_NC_CRON`) или в «реальном времени» в браузере (`nc_search::INDEXING_BROWSER`).

Если ваша служба поиска имеет в своём составе HTTP-робот, который скачивает информацию с сайта, в этом методе необходимо послать команду на запуск переиндексирования. Пути страниц, описанных в \$area_string можно получить следующим образом:

```
$area = new nc_search_area($area_string);
```

```
$urls = $area->get_urls(); // массив только со стартовыми URL
```

Механизм определения того, входит ли страница в заданную область, придётся реализовывать самостоятельно средствами поисковой службы. Если HTTP-робота в службе поиска нет, используйте имеющийся в модуле поиска класс nc_search_indexer:

```
return nc_search_indexer::index_area($area_string, $runner_type);
```

- public function **schedule_indexing**(string \$area_string, int \$timestamp): void
Запланировать переиндексирование области или правила \$area_string в указанное время (\$timestamp — UNIX timestamp). Если вы используете индексатор, встроенный в модуль поиска:

```
return nc_search_scheduler::schedule_indexing($area_string, $timestamp);
```

- public function **is_reindexing**(): mixed

Проверка на то, запущен ли в данный момент процесс индексирования. (Запуск нескольких задач индексирования одновременно невозможен, поскольку может привести к повреждению поискового индекса.)

Метод должен вернуть FALSE, если индексирование в момент вызова не запущено, иначе — объект nc_search_indexer_task в случае использования встроенного индексатора, или TRUE:

```
return nc_search_indexer::get_current_task();
```

- public function **suggest_titles**(string \$input, string \$language, int \$site_id): array

Получить массив с заголовками страниц, подходящих под запрос пользователя (\$input).

Метод должен вернуть двумерный массив с результатами или пустой массив, если подходящих заголовков не найдено. Каждый результат должен содержать элементы "label" (заголовок страницы) и "url" (путь до страницы).

- public function **has_term**(string \$term): boolean

TRUE, если слово имеется в поисковом индексе, иначе FALSE.

- public function **count_documents**(): integer

Получить количество проиндексированных документов.

- public function **count_terms**(): integer

Получить количество различных слов в поисковом индексе.

Желательно, чтобы ваш класс-адаптер мог использовать стандартный для модуля поиска синтаксис запросов. (Поисковый сервер *Solr* полностью совместим не только с языком запросов, но и с форматом файлов,

используемых *Zend_Search_Lucene*, поскольку использует Java-библиотеку *Lucene*.)

Если вам необходима дополнительная информация о классах `nc_search_document`, `nc_search_query`, `nc_search_area`, вы можете узнать подробности из комментариев к их исходным кодам.

После создания класса-адаптера для поисковой службы необходимо добавить в таблицу `Search_Extension` запись о нём:

```
INSERT INTO Search_Extension SET  
ExtensionInterface='nc_search_provider', ExtensionClass='имя вашего  
класса';
```

На вкладке «Настройки» — «Системные» в поле «SearchProvider» укажите имя вашего класса.

Затем, исходя из возможностей службы поиска, необходимо на вкладке «Настройки» — «Расширения» указать расширения, используемые при обработке текста.

Статистика посещений

Модуль «Статистика посещений» предоставляет статистические отчеты, созданные на основе постоянно сохраняемой информации о посетителях и посещенных страницах в ходе эксплуатации сайта. Отчеты представлены как в табличном, так и в графическом виде.

Установка

Установка модуля потребует выбрать родительский раздел, куда будут помещены все разделы модуля. В процессе установки в «Управление задачами» будет так же помещен скрипт `cron.php`.

Настройка

В процессе установки в таблицу управления задачами был помещен скрипт «`/netcat/modules/stats/cron.php?param=test`». Он предназначен для генерации статистики по расписанию. Данный скрипт очень важен, если у вашего сайта большая посещаемость и вы реже, чем раз в день, заглядываете в статистику. Для его работы необходимо по FTP отредактировать файл

cron.php, в самом начале файла удалить строку с «exit;», после чего чуть ниже в строке «\$check="test";» вместо test задать ключевое слово, лучше латинскими буквами. Слово нужно для предотвращения несанкционированного доступа к скрипту. Далее в разделе «Управление задачами» системы администрирования отредактируйте ссылку «/netcat/modules/stats/cron.php?param=test», указав вместо слова test выбранное вами ранее ключевое слово.

Описание настройки функционала «Управление задачами» находится в одноименном разделе данной документации.

Модуль содержит следующие отчеты:

- сводная статистика – суммарная статистика посещений за сегодняшний, вчерашний день, неделю и месяц.
- посещаемость – подробный отчет о посещаемости сайта за указанный период времени с точностью до часа, а также отчет о средней посещаемости сайта.
- популярность страниц – список наиболее посещаемых страниц сайта.
- ссылающиеся страницы – список страниц, с которых посетители приходят на сайт.
- операционные системы – отчет об операционных системах, которыми пользуются посетители сайта.
- браузеры – отчет о браузерах, которыми пользуются посетители сайта.
- IP-адреса – список наиболее активных IP-адресов с возможностью просмотреть информацию по каждому из них.
- география – статистика по странам;
- поисковые фразы – фразы, которые пользователи набирали в поисковых системах, после чего по ссылке с результатов поиска зашли на ваш сайт.

Все отчеты вы можете просмотреть в разделе «Инструменты – Модули» в специальном интерфейсе модуля «Статистика посещений».

В блоке «Популярность страниц» при генерации статистики домены example.net и www.example.net уравниваются. На странице отображаются ссылки с доменом без www, однако они включают в себя и ссылки с доменом с www. По сути, example.net и www.example.net считаются алиасами и рассматриваются как один домен.

Для уменьшения объема БД статистику можно удалять, а именно очистить до какой-то даты. Форма для удаления находится внизу страницы сводной статистики.

Используемые функции:

- **stats_Log()**

Функция регистрирует различную информацию о посетителе сайта в таблицу Stats_Log.

- **stats_CreateReports ()**

Функция создает всевозможные отчеты по статистике. Вы можете создавать новые отчеты: для этого необходимо создать функцию, которая будет генерировать необходимый отчет (по аналогии с функциями stats_CreateReportAttendance, stats_CreateReportPopularity, stats_CreateReportReferer, stats_CreateReportIP, stats_CreateReportBrowser, stats_CreateReportOS) и поместить ее в stats_CreateReports.

- **stats_CreateReports...()**

Функции выводят отчеты на страницу.

Данный модуль не требует настройки параметров. Фактически константа только одна – INSTALLED. Установка ее в ноль «выключит» сбор статистики для проекта.

Управление подписками и рассылками

Общее описание

Модуль предназначен для уведомления пользователей о различных событиях, произошедших на сайте.

Установка

Установка модуля осуществляется через систему администрирования в разделе «Установка модуля».

После успешной установки вам будет предложено создать раздел, в котором пользователи смогут управлять своими подписками. В этом разделе будет автоматически создан соответствующий компонент.

Требования

Для работы модуля необходимы:

- работающая функция mail (ваш хостинг-провайдер должен разрешать отправку писем через сайт)
- настроенный крон (скрипт `/netcat/admin/crontab.php` должен быть занесен в `crontab` Вашего хостинг-провайдера. Подробнее в «Руководстве разработчика»);
- правильное заполнение полей «домен» и «зеркала» в настройках сайта .

Для того, чтобы вам были доступны все возможности модуля «Подписки и рассылки», желательно также иметь модуль «Интерфейс пользователя». В этом случае вы сможете организовывать рассылки и для незарегистрированных пользователей.

Типы рассылок

Модуль создает три типа рассылок:

- уведомления об обновлениях страниц;
- регулярная рассылка;
- сервисная рассылка.

Рассмотрим каждый тип рассылок подробнее.

Подписка на раздел

Данный тип рассылки позволяет уведомлять пользователей об обновлениях раздела. Это могут быть как свежая новость или новая вакансия, так и пост в блоге и прочее. Каждую рассылку такого типа необходимо связать с разделом, а точнее с компонентом в разделе. В этом случае все подписавшиеся пользователи будут получать рассылку именно из этого раздела. Пользователь может выбрать себе период получения писем (естественно, если за заданный период времени никаких событий не произошло, то пользователь просто не получит письмо).

Письмо формируется на основе шаблона и добавленных и/или изменённых (в зависимости от настроек рассылки) объектов.

Рассмотрим этот простой алгоритм на примере рассылки «Новости». Для начала нужно создать новую рассылку (о работе с рассылками рассказано ниже, в главе «Управление рассылками и подписчиками»), придумать ей название, затем определить типы пользователей, которые смогут подписаться на эту рассылку (все, зарегистрированные или только уполномоченные). Далее необходимо задать компонент в том разделе, информация об обновлении которого и попадет в рассылку. В нашем примере это компонент в разделе «Новости». Теперь нужно задать шаблон письма, по которому будут формироваться письма для подписчиков. К примеру, шаблон письма может содержать анонс новости и ссылку для полного просмотра, а также полезную информацию (ссылку для перехода на страницу управления собственными подписками и прочее). После создания рассылки пользователи смогут на неё подписаться (пройдя по определённой ссылке или через страницу управления подписками). Спустя некоторое время администратор сайта добавит новость, и каждому подписавшемуся на эту рассылку пользователю придёт электронное письмо, содержащее анонс этой новости и ссылку для полного просмотра. В зависимости от заданного пользователем периода получения писем, он может получать в одном письме уведомления о нескольких новостях. Например, если пользователь подписан на уведомления с периодом «раз в неделю», то ему будет приходить одно письмо в неделю, содержащее все новости, которые были добавлены за этот период.

Пользователь может подписаться не только на раздел в целом, но и на конкретный объект в частности. В этом случае он будет получать только информацию об этом объекте (например, новые комментарии к выбранному объекту).

Выпуск по этой рассылке формируются автоматически для каждого пользователя.

Регулярная рассылка

Данная рассылка отправляется всем подписавшимся на неё пользователям с заранее утверждённым шаблоном письма и с той периодичностью, которая задана администратором сайта.

Например, рассылка может содержать скомпилированную информацию за последнюю неделю. В одном письме пользователь получит новости, вакансии и прочее. Для примера рассмотрим еженедельную рассылку, которая содержит новые товары, список актуальных вакансий и рекламную информацию. Для этого необходимо реализовать вывод подходящих товаров и вакансий. Удобнее всего это сделать через «шаблоны компонента». Естественно, сначала необходимо создать рассылку, выбрать тип «регулярная рассылка», задать периодичность, например, «один раз в неделю». В шаблоне письма необходимо прописать вывод товаров и вакансий, а также разместить рекламу (например, ссылку с картинкой). Раз в неделю подписанные на эту рассылку пользователи будут получать такое письмо.

Сервисная рассылка

Письма этой рассылки отправляются всем пользователям сайта, тип которых подходит под заданные администратором условия. С помощью данного типа рассылки вы сможете уведомлять пользователей об истечении баланса, поздравлять их с праздниками и прочее.

Управление рассылками и подписчиками

Список рассылок

В управлении модулем можно увидеть все существующие рассылки. Данные выводятся в виде таблицы. Столбцы у этой таблицы такие:

- ID — номер рассылки;
- название рассылки;
- раздел (заполняется только для рассылок типа «подписка на раздел») — раздел, из которого берется информация для рассылки;
- тип — тип рассылки;
- подписчики — количество подписчиков. Если подписчики существуют, то можно кликнуть на ссылку, чтобы увидеть всех подписавшихся;
- настройки — переход к настройкам раздела;

В последнем столбце можно выбрать необходимые рассылки, чтобы произвести с ними операции, такие как:

- удаление;
- включение - рассылки становятся активными;
- выключение - рассылки становятся неактивными.

Рассылки серого цвета в таблице — неактивные.

Для удобства в списке рассылок существуют выборки по типу (подписка на раздел, регулярная рассылка, сервисная рассылка) и по статусу (активная, неактивная).

Создание и изменение рассылки

Для создания новой рассылки необходимо перейти в управление модулем «Подписки и рассылки» в административной части сайта. Внизу справа вы увидите кнопку «Добавить рассылку».

Для изменения рассылки кликните на шестерёнку в списке.

При добавлении (изменении) рассылки необходимо заполнить (изменить) следующие поля:

- название — название рассылки;
- тип рассылки — подписка на раздел, регулярная рассылка или сервисная рассылка;
- тип доступа — типы пользователей, которые смогут подписаться на данную рассылку;
- активна — будут ли рассылаться письма от этой рассылки;
- сохранять подробную статистику — данная опция позволяет вести подробную статистику по данной рассылке (она включает всю информацию о подписках, рассылках и отписках от данной рассылки в формате «пользователь/время»);
- условие подписки — заданное условие будет выполняться перед подпиской, и, в случае необходимости, подписку можно отменить;
- условие рассылки — данное условие будет выполняться перед отправкой писем подписчикам;
- действие после подписки — заданное действие будет выполняться после подписки;
- тема письма;
- HTML-письмо — отправлять письмо как HTML;
- хэдер, основной контент и футер — шаблон письма. Подробнее эту опцию рассмотрим далее, в главе «Принцип формирования письма».

Для разных типов рассылок существуют и специфические настройки.

Для рассылок типа «подписка на раздел» необходимо указать сайт, раздел и компонент в разделе, на который осуществляется подписка, а также тип действия, после которого объект попадет в рассылку. Это может быть:

- добавление включённого;
- добавление выключенного;
- изменение включённого;
- изменение выключенного;
- включение;
- выключение.

Для регулярной и сервисной рассылки необходимо указать период рассылки.

Список подписчиков

Для рассылок типа «подписка на раздел» и «регулярная рассылка» можно увидеть список подписавшихся пользователей, где также существует возможность управления подписками. Для этого в списке рассылок нужно кликнуть на число подписчиков.

Таблица подписчиков на рассылку содержит три столбца: пользователь, статус и выбор подписчика.

Статус может принимать следующие значения:

- включён — пользователь принимает письма;
- ожидание подтверждения — пользователю выслано письмо с ссылкой для активации рассылки;
- выключен — пользователю письма по данной рассылке не отсылаются.

С выбранными подписчиками можно произвести следующие действия:

- удалить — отписывает пользователя от рассылки;
- выключить;
- включить.

Подписки пользователя

После установки модуля «Подписки и рассылки» в системе администрирования при редактировании пользователя появится новая вкладка — «Подписки пользователя».

В этой вкладке Вы можете увидеть все подписки пользователя, статус этих подписок, а также удалить, выключить, включить или добавить новую рассылку (подписать пользователя).

Статистика рассылок

Для каждой рассылки выдается статистика, которая включает количество высланных писем и время последней рассылки. Увидеть статистику можно, кликнув на вкладку «Статистика» в управлении модуля.

Если рассылка содержит подробную статистику (включать/выключать эту возможность можно в настройках модуля), то Вы можете перейти к этой информации, кликнув на название рассылки при показе статистики.

Подробная статистика выводится в виде таблицы со следующими столбцами:

- пользователь;
- действие (подписка, подтверждение подписки, отписка, отправка письма);
- дата и время.

Единоразовая рассылка

Модуль позволяет производить единоразовую рассылку по базе пользователей. Для использования этой возможности вам нужно выбрать пользователей, которые получают письмо, и сформировать письмо.

Пользователей можно выбирать по следующим критериям:

- группа, в которую он входит;
- подписчики определенной рассылке;
- тип пользователя (зарегистрированные или все пользователи в базе);
- активность пользователя (включенные или выключенные);
- дополнительные поля из системной таблицы «Пользователи».

В шаблоне письма необходимо указать тему письма и сам текст письма. Текст письма можно редактировать в визуальном редакторе. При желании можно прикрепить к письму до трех файлов.

Сформированное письмо для проверки можно отправить на свой адрес. Для этого нужно нажать на кнопку «Послать тестово» в нижней части рабочей области. Адрес для тестовой рассылки задается в настройках модуля (см. следующую главу).

Настройки модуля

В этой секции задаются основные параметры модуля:

- разрешить объединение писем из различных разделов — при включении этой опции письма рассылки типа «подписка на раздел» будут объединяться в одну. Таким образом, если пользователь подписан на разделы «Новости» и «Блог», при совпадении отправки

писем по этим рассылкам, ему придёт не два отдельных, а одно совмещенное письмо;

- имя отправителя — имя отправителя рассылки;
- e-mail отправителя — E-mail отправителя рассылки;
- обратный адрес — обратный электронный адрес отправителя рассылки;
- тестовый адрес для рассылок — это адрес будет использоваться для тестовых рассылок;
- кодировка письма — кодировка отправляемых писем;
- поле из таблицы "Пользователи" с E-mail — название поля из таблицы «Пользователи», на которое будут отправляться письма. Поле должно иметь тип «строка» с форматом «email»;
- подтверждение подписки — определяет, в каких случаях требовать подтверждение подписки. Возможные значения:
 - только незарегистрированным — требовать подтверждение только от незарегистрированных пользователей;
 - при первой подписке — требовать подтверждение только в случае, если пользователь не подписан ни на одну рассылку;
 - всегда — подтверждать подписку пользователю нужно будет каждый раз.
- удалять неподтверждённую подписку через — задает время в часах, после которого неподтвержденная подписка будет удалена;
- тема письма — тема письма о подтверждении подписки;
- HTML-письмо — если включена эта опция, то письмо для подтверждения подписки отправится как HTML;
- тело письма — шаблон письма для подтверждения подписки;
- форма подписки для незарегистрированного пользователя — HTML-код формы, который будет выводиться при подписке на рассылку незарегистрированным пользователям. Обычно содержит элемент ввода электронного адреса;
- текст, выводимый при подтверждении подписки — данный текст будет показан пользователю после перехода по ссылке для подтверждения подписки;
- текст, выводимый при отписке — данный текст будет показан пользователю при отказе от получения рассылки;
- текст, выводимый при ошибке — данный текст будет показан пользователю при возникновении ошибки, связанной с подтверждением подписки. Например, срок ожидания подтверждения уже истек.

Доступы на рассылку и назначение прав

Для каждой рассылки можно назначить тип доступа:

- всем — любой пользователь сможет подписаться на рассылку;
- зарегистрированным — подписаться на рассылку смогут только зарегистрированные пользователи;
- уполномоченным — подписаться смогут только уполномоченные пользователи, а это:
 - директор;
 - супервизор;
 - подписчик (для такого типа прав необходимо задать рассылку, на которую пользователь сможет подписаться).

Права задаются через систему администрирования в разделе «Пользователи».

Для того чтобы незарегистрированный пользователь смог подписаться на рассылку, Вам необходимо иметь модуль «Интерфейс пользователя».

Принцип формирования писем

Все письмо можно разделить на три части: хэдер, содержательная (контентная) часть и футер.

Для рассылок типа «подписка на раздел» содержательная часть будет повторяться для каждого объекта, вошедшего в рассылку для пользователя.

Допустим, пользователь подписан на рассылку «Новости» с периодичностью 2 часа, и за эти два часа администратор добавил три новости, - тогда письмо для этого пользователя схематично будет выглядеть так:

Хэдер
Контентная часть для 1-ой новости
Контентная часть для 2-ой новости
Контентная часть для 3-ой новости
Футер

Для регулярных и сервисных рассылок схема письма такая:

Хэдер
Контентная часть

В шаблонах писем через определённые переменные доступна информация о пользователе, сайте, разделе, компоненте в разделе и объектах.

Шаблоны писем, условия и отправка писем

Шаблоны писем

Шаблоны писем (хэдер, содержательная часть, футер) — это обычный html-код с возможностью вставки php-кода.

Спецсимволы в теме письма, хэдере, контентной части футере нужно экранировать.

Для вставки функций можно воспользоваться следующей конструкцией:

html-код".some_function()."html-код

Возвращаемое значение этой функции войдет в письмо.

Для вставки php-кода можно применить следующую схему:

```
HTML-код
";
php-код
$result .= "
```

Каждая часть шаблона письма (условие, хэдер, футер, основная часть) проходит обработку функцией eval. Если в шаблоне отсутствует использование массива \$current_user, то обработка будет производиться только один раз для всех пользователей.

Список доступных переменных приведён ниже.

Условие подписки

Перед подпиской пользователя на рассылку выполняется условие подписки. В условие подписки нужно писать «чистый» php-код. Если после выполнения этого кода переменная \$posting станет равной 0, то подписки не произойдет, а пользователю будет выведен текст, содержащейся в переменной \$warnText.

Условия рассылок и отправка писем

Перед формированием письма для каждого пользователя выполняется условие рассылки. Если по окончании выполнения условия рассылки переменная `$posting` окажется равной 0, то письмо не будет отправлено.

Пример:

отправлять письма только тем пользователям, у которых значение `Rating` больше 100.

В условии рассылки:

```
if ( $current_user['Rating'] <= 100 ) {  
    $posting = 0;  
}
```

Если в хэпере, футере или содержательной части письма рассылки «подписка на раздел» переменная `$mail_not_send` примет значение 1, то рассылка не будет производиться.

Список доступных переменных и макропеременных

Для выполнения условий, получение хэдера, футера и содержательной части письма каждый раз вызывается функция, поэтому переменные, объявленные в, допустим, условии рассылки недоступны в футере письма. Чтобы все-таки передавать между ними некоторые значения, можно воспользоваться заранее определённым статическим массивом `$static_vars`.

Например, в условие рассылки пишем:

```
$static_vars['my_var'] = 123;
```

тогда в футере будет доступен элемент массива `$static_vars['my_var']` со значением 123.

Условие рассылки

`$current_user` — массив с параметрами пользователя, которому отправляется письмо.

Ключи этого массива таковы:

- `User_ID` — номер пользователя;
- `PermissionGroup_ID` — главная группа пользователя;
- `Permission_Group` — массив со номерами всех групп пользователя;

NetCat 4.5 Руководство по модулям

- Checked — пользователь включён или выключен;
- Language — язык пользователя;
- Created — дата и время регистрации пользователя;
- LastUpdated — дата и время последнего изменения пользователя;
- Auth_Hash — ключ для авторизации по хэшу.

Также массив содержит все значения дополнительных полей. Поля задаются в системной таблице «Пользователи». Ключ массива в этом случае соответствует имени поля (если в системной таблице «Пользователи» есть поле целого типа с латинскими именем Rating, то в условии подписки будет доступен элемент массива `$current_user['Rating']`, содержащий значение этого поля).

Для некоторых типов полей будут доступны еще некоторые элементы.

Для полей типа «Список»и «Множественный список» доступны идентификаторы выбранных элементов через `$current_user['ИмяПоля_id']`.

Для поля типа «Файл»:

`$current_user['ИмяПоля_name']` – оригинальное имя файла;

`$current_user['ИмяПоля_type']` – тип файла;

`$current_user['ИмяПоля_size']` – размер файла;

`$current_user['ИмяПоля_url']` – реальный путь до файла.

`$mailer` — массив с информации о рассылке.

Действие после подписки

`$current_user` — массив с параметрами пользователя, которому отправляется письмо;

`$current_catalogue`, `$current_sub`, `$current_cc` (для подписок на раздел);

`$mailer` — массив с информации о рассылки;

`$nc_status` — статус подписки ('on' — включена, 'wait' — требуется подтверждение).

Тема письма

`%USER_LOGIN%` — логин пользователя;

`%LINK%` — ссылка для отказа от рассылки;

`%SUB_LINK%` — ссылка на страницу управления подписками;

`$current_user` — массив с параметрами пользователя, которому отправляется письмо;
`$current_catalogue`, `$current_sub`, `$current_cc` (для подписок на раздел);
`$mailer` — массив с информацией о рассылке.

Хэдер и футер

`%USER_LOGIN%` — логин пользователя;
`%LINK%` — ссылка для отказа от рассылки;
`%SUB_LINK%` — ссылка на страницу управления подписками;
`$current_user` — массив с параметрами пользователя, которому отправляется письмо;
`$current_catalogue`, `$current_sub`, `$current_cc` (для подписок на раздел);
`$classID` — номер компонента (для подписок на раздел);
`$mailer` — массив с информацией о рассылке.

Основная часть

`%USER_LOGIN%` — логин пользователя;
`%LINK%` — ссылка для отказа от рассылки;
`%SUB_LINK%` — ссылка на страницу управления подписками;

`$current_user` — массив с параметрами пользователя, которому отправляется письмо;
`$current_catalogue`, `$current_sub`, `$current_cc` (для подписок на раздел);
`$classID` — номер компонента (для подписок на раздел);
`$mailer` — массив с информацией о рассылке;
`$f_RowID` — номер объекта;
`$f_UserID` — ном пользователя, добавившего объект.
`$f_ИмяПоля` — значение поля объекта.

Использование `$f_ИмяПоля` аналогично использованию такой же переменной в компонентах. Допустим есть рассылка новостей, а в компоненте «Новости» используются поля `Title` (для заголовка), `Announce` (для анонса). Тогда шаблон для основной части может выглядеть так:

```
<a href='http://mysite.ru".nc_message_link( $f_RowID,  
$classID) .' '>  
    $f_Title  
</a>  
<br/>  
$f_Announce  
<br/><br/>
```

Подтверждение подписки

%USER_LOGIN% — логин пользователя;

%LINK% — ссылка для подтверждения;

%SUB_LINK% — ссылка на страницу управления подписками;

Служебные переменные

Список переменных, которые нельзя переопределять в условиях и шаблонах рассылок:

\$item

\$cache

Периоды получения писем

Все периоды, с которыми идет работа, задаются в списке «Периоды подписки». Латинское название этого списка должно быть **SubscriberPeriod**.

Каждый элемент этого списка — отдельный период. В дополнительном значении элемента задается числовое значение периода в секундах. Например, для периода «каждые 2 часа» дополнительное значение должно быть $2*60*60 = 7200$.

Дополнительное значение также может быть равным нулю.

Способы подписки

Существуют два способа подписки на рассылку:

1. Через ссылку вида `example.com/news/subscribe_news.html`

2. Передавая необходимые данные на скрипт `/netcat/modules/subscriber/index.php`

В первом случае подписаться можно только на одну рассылку, принадлежащую компоненту в разделе, в указанном примере — это компонент в разделе `news` из раздела `news`. Ссылка доступна в компоненте (в префиксе, в объекте списка, в суффиксе, в полном отображении объекта) через переменную `$subscribeLink`, то есть, например, в префиксе компонента можно написать:

`<a href=" ".$subscribeLink, ".">подписаться</a>`

Это код выведет ссылку для подписки на текущий компонент в разделе.

Формат ссылки в общем виде:

`http://название_сайта/раздел1/.../разделN/subscribe_КлючевоеСловоКомпоне
нтаВРазделе.html`

При заходе по такой ссылке неавторизованный пользователь увидит форму подписки для неавторизованного пользователя, которая задается в настройках модуля.

Ссылку вида `example.com/news/subscribe_news.html` можно использовать и для отписки пользователя от рассылки. Для этого в условие подписки нужно проверять, подписан ли пользователь на рассылку, и, если подписан, то отписать его. Пример такой реализации:

```
// получаем объект для работы с рассылками
$nc_subscriber = nc_subscriber::get_object();
// проверяем, подписан ли пользователь на компонент в разделе
if ( $nc_subscriber->is_subscribe_to_cc($cc) ) {
    // получаем номер рассылки по данному компоненту в разделе
    $mailer_id = $nc_subscriber->get_mailer_by_cc($cc,
'Mailer_ID');
    // отписываем пользователя
    $nc_subscriber->subscription_delete_by_mailer($mailer_id);
    // подписка производиться не будет
    $posting = 0;
    // текст, выдаваемый пользователю
    $warnText = "Вы успешно отписались";
}
```

Подробнее о функция модуля будет рассказано в специальном разделе.

Во втором случае формат данных, пришедших через `get` или `post`, должен быть таким:

`subscribe_НомерРассылки=1` — для подписки на рассылку;

`subscribe_НомерРассылки=-1` — для отписки от рассылки.

Пример:

Ссылка для подписки на рассылки с номерами 2 и 5, а также для отписки от рассылки с номером 3:

`http://example.com/netcat/modules/subscriber/index.php?`

`subscribe_2=1&subscribe_5=1&subscribe_3=-1`

Структура таблиц

Ниже приведен список таблиц, которые используются модулем.

Subscriber_Mailer — таблица хранит все рассылки. Основные поля:

- Mailer_ID — номер рассылки;
- Name — имя рассылки;
- Type — тип рассылки (1,2 или 3);
- Access — тип доступа к рассылки;
- Sub_Class_ID — номер компонента в разделе, к которому привязана рассылка (для рассылок типа «подписка на раздел»).

Subscriber_Subscription — таблица хранит подписки пользователя на рассылки. Основные поля:

- Mailer_ID — номер рассылки;
- User_ID — номер пользователя;
- Period — номер периода.

Subscriber_Log — в этой таблице хранится полная статистика по рассылке.

Subscriber_Message — таблица хранит номер объектов, которые попадут в рассылку. Объект хранится через связку (Sub_Class_ID, Message_ID). В таблице так же присутствуют поля Rel_Sub_Class_ID и Rel_Message_ID. Они определяют тот объект, на который пользователь должен быть подписан, чтобы получить в рассылку объект (Sub_Class_ID, Message_ID). Записи этой таблицы автоматически очищаются, если со времени добавления прошло больше времени, чем максимальное значение периода в элементах списка «Периоды подписки».

Subscriber_UserMessage — таблица хранит подписки пользователя на объекты.

Subscriber_Prepared — таблица хранит готовые письма, которые отправятся при очередном запуске скрипта.

Subscriber_Settings — таблица хранит настройки модуля.

Компонент «Список подписок»

Компонент «Подписки и рассылки: Список подписок», входящий в состав модуля, используется для управления пользователем собственными подписками. С помощью него пользователь сможет подписаться и отписаться от рассылок. С помощью визуальных настроек компонента можно настроить вывод рассылок на страницу.

Основной запрос формируется в системных настройках. С помощью переменной \$ignore_all данные «тянутся» из таблицы Subscriber_Mailer. Так же в запросе подключается таблицы Subscriber_Subscription, для того, чтобы узнать существующие подписки пользователя, Classifier_SubscriberPeriod — для получения периодов, Sub_Class — для получения информации о компоненте в разделе.

Вы сами можете модифицировать запрос (см. «Руководство разработчика» - «Системные настройки компонента») для удовлетворения собственных нужд (например, выборка рассылок только по определённому сайту, и т.д.).

Данные с формы, выводимой этим компонентом, попадают на скрипт /netcat/modules/subscriber/index.php

Раздел, к которому прикручен данный компонент, будет использоваться при подтверждении подписки.

Подписка на объект

Модуль позволяет пользователю подписываться на объект (а не на компонент в разделе в целом). В этом случае пользователь будут получать уведомления о событиях, произошедших с данным объектом или с другими объектами, логически связанными с ним. Приведем простой пример: на сайте имеется блог, к которому сообщениям которого пользователи могут добавлять комментарии, необходимо, чтобы читатели смогли отслеживать комментарии по определённому посту. Если просто организовать рассылку по компоненту с комментариями, то подписанные пользователи будут получать все новые комментарии (вне зависимости от того, к какому посту принадлежит комментарий), что неприемлемо, но мы может позволить пользователям подписаться на конкретный пост (на конкретный объект) и при добавлении нового комментария к посту вставлять объект с новоиспечённым комментарием в очередь рассылок для данного пользователя. Логическую связь между объектами и условием добавления объекта в рассылку определяет сам разработчик. Заметим, что подписка на объект относится к типу рассылок «подписка на раздел».

Рассмотрим пример реализации задачи, описанной выше.

Для определённости будем считать, что у нас есть компоненты «Блог», «Комментарии к блогу» (у этого компонента есть поля «Text» - текст комментария, «Post_ID» - номер поста из компонента «Блог», к которому относится комментарий), а так же созданы разделы с этими компонентами.

Для начала нужно создать рассылку, назовем её «Комментарии к блогу», тип рассылки - «подписка на раздел», в специфических настройках нужно указать номер компонента с блогом (ведь подписка идет на объекты именно этого компонента), параметр «Добавлять объект в рассылку при» нужно оставить пустым, мы сами будем вызывать функцию для добавления объекта в рассылку. Для простоты заполним содержательную часть письма следующим кодом:

```
Новый комментарий:<br/>
$f_Text
<br/>
```

Вы так же можете заполнить остальные поля из шаблона письма и условий рассылки.

Далее необходимо вывести ссылку для подписки на объект. Ссылка на подписку на объект выглядит следующим образом:

```
http://название_сайта/раздел1/.../разделN/subscribe_ИмяКомпонентаВРазделе_НомерОбъекта.html
```

Такая ссылка доступна через переменную **\$subscribeMessageLink** в объекте в списке компонента, а так же в отображении объекта. Итак, в шаблоне отображения объекта на отдельной странице компонента «Блог» пишем:

```
<a href='$subscribeMessageLink'>Отслеживать комментарии</a>
```

И наконец, необходимо добавлять объект с комментарием в очередь рассылок. В действие после добавления объекта в компоненте «Комментарии к блогу» нужно прописать следующий код:

```
";
// получаем объект для управления рассылками
$nc_subscriber = nc_subscriber::get_object();
// добавляем объект в очередь рассылок
$nc_subscriber->new_message ( $cc, $message, XX, $f_Post_ID );
echo "
```

Здесь: **\$cc**, **\$message** — номер компонента в разделе с комментариями и номер добавленного объекта (комментария) соответственно:

XX — номер компонента в разделе с компонентом «Блог».

О методе **new_message** рассказано ниже.

Интеграция с модулем «Форум2»

Данный модуль можно использовать для организации подписок на топики форума, работающего под управлением модуля «Форум 2». Для каждого нового форума создаётся новый раздел, поэтому необходимо создавать новую рассылку. Модуль «Форум2» при создании нового форума сам может создать рассылку на основе имеющегося (по существу, скопировав все параметры «шаблонной» рассылки). Шаблонная рассылка появляется при установке модуля «Подписки и рассылки» и имеет название «Форум: Шаблон рассылки». Это рассылка выключена. Номер этой рассылки нужно указать в настройках модуля «Форум2», параметр — SUBSCRIBER_MAILER_ID.

Обратите внимание, что при подписке на топик идёт подписка на объект.

Совместимость с предыдущими версиями

Для совместимости с предыдущими версиями (до версии 4.0 исключительно) модуль может работать в «старом» режиме, то есть используется те же функции, что и в версиях до 4.0, а так же используется только одна таблица — Subscriber. Функционал, описанный в данном руководстве при использовании такого режима недоступен. Руководство по «старому» модулю можно скачать на сайте netcat.ru

При установке параметра VERSION в значение 2 (по умолчанию) модуль функционирует в нормальном режиме.

Организация подписки для незарегистрированных пользователей.

При наличии модуля «Интерфейс пользователя» можно разрешить незарегистрированному пользователю подписываться на рассылки. Хотя пользователь и не регистрируется на сайте, он все равно создаётся (добавляется запись в таблице User). Назовем таких пользователей псевдопользователи.

Для начала необходимо настроить модуль «Интерфейс пользователя», добавив следующие параметры и их значения в настройки модуля:

PSEUDOUSERS_ENABLED 1 - разрешаем псевдопользователей.

PSEUDOUSERS_GROUP НомерГруппы — номер группы, куда попадут псевдопользователи.

PSEUDOUSERS_FIELD ИмяПоля — имя поля, которое будет уникально для каждого псевдопользователя. В данном случае лучше указать Email.

NetCat 4.5 Руководство по модулям

Псевдопользователь сможет авторизоваться только в одном разделе (разделе с управлениями подписками), на других разделах он будет как неавторизованный, причем для авторизации он должен пройти по ссылке, содержащую специальный код в параметре `auth_hash`. Такая ссылка генерируется автоматически и доступна через макропеременную `%SUB_LINK%`.

Допустим, мы хотим разрешить всем пользователям подписываться на новости компании. Для этого создаем рассылку «Новости» с типом доступа «всем».

В настройках модуля необходимо задать форму подписки для неавторизованных пользователей. Она может выглядеть так:

```
Подписка на новости! <br/>
<form method='post' action='/netcat/subscribe.php'>
  <input type='hidden' name='catalogue' value='".
$catalogue.'" />
  <input type='hidden' name='sub' value='". $sub.'" />
  <input type='hidden' name='cc' value='". $cc.'" />
  <input type='hidden' name='posting' value='1' />
  Ваш email:
  <input type='text' name='fields[Email]' value='".
$fields[Email].'" />
  <input type='submit' value='Ok' />
</form>
```

Основные моменты:

1. Данные передаются на скрипт `/netcat/subscribe.php`
2. Необходимо передать номер сайта, раздела и компонента в разделе.
3. Необходимо передать параметр `posting`, равный 1.
4. Необходимо передать параметр `fields[ИмяПоля]`. Значение данного параметра запишется в таблицу `User`, в поле `ИмяПоля`. Рекомендуется использовать `Email`, иначе пользователь не сможет получать письма.

Если пользователь не ошибся при вводе адреса, то далее сработает «Действие после подписки», которое может содержать текст об успешной подписке. Пример заполнения действия после подписки:

```
".( $nc_status == 'on' ? "Вы успешно подписались." : "На ваш e-
mail отправлено письмо с ссылкой для подтверждения подписки")."
<br/><br/>
<a href='". $current_sub['Hidden_URL'].'">Вернуться к
новостям</a>
```

Пользователю придет письмо, в котором будет ссылка на подтверждение подписки. Шаблон письма задается в настройках модуля. Пример шаблона письма:

Уважаемый ...!

Чтобы подтвердить подписку, перейдите по `<a href='%LINK%'>ссылке.</a>`
`<a href='%SUB_LINK%'>Страница управления подписками</a>`

В настройках модуля так же можно задать максимальное время ожидания подтверждения подписки.

При переходе по ссылке для подтверждения пользователь увидит текст, заданный в настройках модуля («Текст, выводимый при подтверждении подписки»).

После этого незарегистрированный пользователь сможет получать обновления о сайте и управлять своими подписками в специальном разделе.

Как было сказано выше, псевдопользователь авторизуется по хэшу в разделе управления собственными подписками, то есть переменные `$AUTH_USER_ID` и `$current_user` доступны.

Элемент массива `$current_user['UserType']` для псевдопользователей равен 'pseudo', для обычных зарегистрированных пользователей — 'normal'.

Константы модуля

В настройках модуля следует задать два параметра:

`SUBSCRIBER_CLASS_ID` — номер компонента «Список подписок».

`VERSION` — версия модуля (по умолчанию 2)

Классы модуля

Основной класс для работы с подписками и рассылками называется **nc_subscriber**.

Для создания или получения объекта этого класса необходимо вызвать статический метод класса **get_object()**:

```
$nc_subscriber = nc_subscriber::get_object();
```

Имя объекта может быть любым. Далее будем считать, что объект определен и имеет имя `$nc_subscriber`.

get_object()

Статический метод. Возвращает объект класса, а при первом обращении инициализирует объект.

Пример использования:

```
$nc_subscriber = nc_subscriber::get_object();  
// теперь можно использовать $nc_subscriber
```

mixed get (int \$mailer_id, string \$item = '')

Метод позволяет получить информацию о рассылке по ее номеру. Если параметр `$item` не задан, то функция возвращает массив с параметрами, иначе — только один параметр.

Ключи возвращаемого массива могут быть такие: Name, Type, Active, Access, InStat, Sub_Class_ID, ActionType, Period, SubscribeCond, SendCond, SubscribeAction, Header, Record, Footer, Subject, HTML. Подробнее о параметрах написано чуть ниже.

Примеры использования:

1. Вывести имя рассылки с номером 7:

```
echo $nc_subscriber->get(7, 'Name');
```

2. Вывести тип доступа к рассылке с номером 7:

```
switch ( $nc_subscriber->get(7, 'Access') ) {  
    case 1:  
        echo «Рассылка доступна всем»;  
        break;  
    case 2:  
        echo «Рассылка доступна только зарегистрированным»;  
        break;  
    case 3:  
        echo «Рассылка доступна только уполномоченным»;  
        break;  
}
```

3. Получить всю информацию о рассылке с номером 7:

```
$mailer = $nc_subscriber->get(7);
```

после выполнения этого кода массив \$mailer будет иметь вид:

array ('Name' => 'Имя рассылки', 'Type' => 1, 'Active' => 1, 'Access' => 2, итд)

mixed get_mailer_by_cc (int \$cc, string \$item = '')

Метод позволяет получить информацию о рассылке, привязанной к компоненту в разделе с номером \$cc. Если рассылки для этого компонента не существует, то будет создано исключение ExceptionMailer. Метод аналогичен предыдущему за тем исключением, что вместо номера рассылки нужно передавать номер компонента в разделе.

Пример использования:

для текущего компонента в разделе вывести (если существует) ссылку для подписки, указав ее имя:

```
try {
    $nc_subscriber = nc_subscriber::get_object();
    $mailer_name    = $nc_subscriber->get_mailer_by_cc($cc,
'Name');
    echo "< a href='". $subscriberLink. "'>". $mailer_name. "</a>";
}
catch ( Exception $e) {
}
```

int add (array \$params)

Метод создает новую рассылку. В качестве параметра нужно передать ассоциированный массив, ключи которого — параметры рассылки (их список будет приведен ниже).

Ключи, которые могут использоваться:

- Name — имя рассылки;
- Type — тип рассылки. Возможные значения:
 - 1 — подписка на раздел (по умолчанию);
 - 2 — регулярная рассылка;
 - 3 — сервисная рассылками.
- Active — активность рассылки (включена или нет). Возможные значения:
 - 0 — рассылка выключена;
 - 1 — рассылка включена (по умолчанию).
- Access — тип доступа. Возможные значения:
 - 1 — доступно всем (по умолчанию);
 - 2 — доступно зарегистрированным;
 - 3 — доступно уполномоченным.
- InStat — ввести ли подробную статистику рассылки. Возможные значения:

- 0 — нет;
- 1 — да.
- Sub_Class_ID — номер компонента в разделе, из которого берется информация для рассылки (для «подписок на раздел»);
- ActionType — числовые обозначения действий, после которых объект попадет в рассылку.
- Period — номер периода (для регулярной и сервисной рассылки);
- SubscribeCond — условие подписки;
- SendCond — условие рассылки;
- SubscribeAction - действие после подписки;
- Header — верхняя часть письма, отправляемой этой рассылкой;
- Record — содержательная часть письма;
- Footer — нижняя часть письма;
- Subject — тема письма;
- HTML — тип письма. Возможные значения:
 - 0 — обычное письмо;
 - 1 — HTML-письмо.

Пример использования.

```
$params = array ('Name' => 'Новая подписка', 'Access' => 2,
'Sub_Class_ID' => 104, 'ActionType' => 1 );
$nc_subscriber->add($params);
```

Данный код создает новую рассылку типа «подписка на раздел» с именем «Новая подписка». Рассылка доступна только зарегистрированным пользователям, информация берется из компонента в разделе с номером 104, а объект добавляется в рассылку при добавлении, причем он должен быть выключенным.

bool update (int \$mailer_id, array \$params = array())

Метод позволяет обновить параметры рассылки. Сами параметры задаются в массиве по аналогии с методом add.

Примеры использования:

1. Поменять тип доступа рассылки с номером 7 на вариант «только уполномоченным»:

```
$nc_subscriber->update(7, array('Access' => 3) );
```

2. Сделать копию рассылки с номером 7, у новой рассылки поменять имя:

```
// получим все параметры рассылки 7 в виде массива
$mailer_old = $nc_subscriber->get(7);
// создадим новую рассылку, метод add вернет ее номер
$mailer_new_id = $nc_subscriber->add($mailer_old);
// поменяем название рассылки
```

```
$nc_subscriber->update( $mailer_new_id, array('Name' => 'Копия рассылки 7') );
```

Обратите внимание, что скопировалось только сама рассылка, но не подписчики.

new_message (int \$cc, int \$message, int \$rel_cc = 0, int \$rel_message = 0)

Функция добавляет объекты с номером \$message из компонента в разделе с номером \$cc в очередь рассылки. Если указаны параметры \$rel_cc и \$rel_message, то пользователь получит в рассылке этот объект только в том случае, если он подписан на объект с номером \$rel_message из компонента в разделе с номером \$rel_cc, в противном случае объект получают все пользователи, подписанные на компонент в разделе \$cc (а точнее, подписанные на рассылку, связанную с компонентом в разделе с номером \$cc).

Пример использования:

допустим, необходимо при добавлении новости выбрать, попадет ли новая информация в рассылку или нет. Для этого в соответствующем компоненте «Новости» создаем новое булево (истина/ложь) поле с именем notify, оно должно быть обязательным для заполнения.

В действии после добавления прописываем следующий код:

```
";
if ( $f_notify ) {
    $nc_subscriber = nc_subscriber::get_object();
    $nc_subscriber->new_message($cc, $message);
}
echo "
```

Теперь при добавлении новости, пользователь через элемент checkbox будет выбирать, отправлять ли извещение о новом объекте или нет. Естественно, в этом примере предполагается, что рассылка для раздела существует.

subscription_add (int \$mailer_id, int \$user_id = 0, int \$period = 0, string \$status = false, array \$fields = array(), int \$message = 0)

Метод подписывает пользователя с номером \$user_id на рассылку \$mailer_id. Если \$user_id не указан, то система берет текущего авторизованного пользователя, а если и его нет — то создает нового псевдопользователя, причем его параметры нужно передать в массиве \$fields (ключи массива — соответствующие поля из таблицы Пользователи). Через \$period можно указать период получения рассылок (для подписок на раздел). Если этот параметр не задан, то будет выбрано значение по умолчанию.

NetCat 4.5 Руководство по модулям

С помощью параметра `$status` можно задать статус подписки. Возможные значения:

- `'on'` — подписка активна;
- `'off'` — подписка неактивна;
- `'wait'` — требуется подтверждение пользователя (пользователь должен перейти по определённой ссылке).

В последнем случае пользователю отправится письмо с ссылкой для подтверждения подписки.

Если `$user_id` не задан или равен 0, а текущий пользователь неавторизован и есть модуль «Интерфейс пользователя», то будет создан новый псевдопользователь, его параметры нужно передать через массив `$fields`.

`$fields` может выглядеть так:

```
array ( 'Email' => 'mail@example.com', 'Name' => 'Nickname')
```

В этом случае пользователь будет иметь электронную почту mail@example.com.

Если нужно подписать пользователя на объект компонента в разделе, а не на весь компонент, то номер этого объекта нужно передать через параметр `$message`.

Пример использования:

подписать текущего пользователя на рассылку 7 и сделать её активной:

```
$nc_subscriber->subscription_add( 7, 0,0, 'on');
```

`subscription_add_by_cc ( int $cc, int $user_id = 0, int $period = 0, string $status = false, array $fields = array(), int $message = 0 )`

Метод позволяет пользователю подписаться на компонент в разделе с номером `$cc`.

Функция аналогична предыдущему за тем исключением, что вместо номера рассылки передается номер компонента в разделе.

Пример использования:

подписать пользователя с номером 3 на компонент в разделе с номером 105:

```
$nc_subscriber->subscription_add_by_cc ( 105, 3,0, 'on');
```

`int activate ( mixed $mailer_ids )`

Метод включает рассылки. Номер рассылки или массив с номерами рассылок передаются через параметр `$mailer_ids`. Функция возвращает количество включённых рассылок.

Пример использования:

включить рассылку с номером 3:

```
$nc_subscriber->active(3);
```

int unactivate (mixed \$mailer_ids)

Метод выключает рассылки. Номер рассылки или массив с номерами рассылок передаются через параметр `$mailer_ids`. Функция возвращает количество выключенных рассылок.

Пример использования:

выключить рассылки с номерами 5 и 7:

```
$nc_subscriber->unactive( array( 5,7) );
```

int delete (mixed \$mailer_ids)

Метод удаляет рассылки. Номер рассылки или массив с номерами рассылок передается через параметр `$mailer_ids`. При удалении рассылки так же удаляются все подписки на эту рассылку, статистика и прочее связанные элементы. Функция возвращает количество удалённых рассылок.

Пример использования:

удалить рассылку с номером 3:

```
$nc_subscriber->delete(3);
```

Личный кабинет

Примечание. До версии 4.5 данный модуль назывался «Интерфейс пользователя».

Модуль позволяет решать следующие задачи:

- регистрация пользователей/изменение регистрационных данных/пароля
- авторизация/деавторизация (завершение сеанса) пользователей на сайте
- просмотр списка пользователей в соответствующем разделе сайта
- отправка личных сообщений
- поиск пользователей, присутствующих на сайте
- восстановление пароля
- система «Друзья/Враги»
- личный счет пользователя и его контактные данные
- интеграция с социальными сетями, OpenID
- авторизация по usb-токену

Кроме этого, возможна также интеграция модуля с другими модулями (например, «Форум», «Минимагазин», «Интернет-Магазин», и т.д.).

Управление и работа с зарегистрированными пользователями производится при помощи интерфейса администрирования в разделе «Пользователи».

Редактирование структуры таблицы «Пользователи», а также настройка шаблонов действий и шаблона отображения пользователей на сайте производится в разделе «Системные таблицы».

Установка и требования

После успешной установки модуля вам будет предложено создать разделы «Регистрация», «Изменение данных», «Изменение пароля» и «Восстановление пароля» (ссылка «завершение установки»). К данным разделам будут автоматически подключены соответствующие компоненты и внешние скрипты.

Для работы модуля необходимо иметь следующие расширения PHP:

- curl (<http://php.net/manual/ru/book.curl.php>)
- json (<http://php.net/manual/en/book.json.php>)

- gmp (<http://php.net/manual/en/book.gmp.php>) — только для авторизации по токенам

Для корректной работы форм авторизации и регистрации требуется js-библиотека jquery (<http://jquery.com/>), которая поставляется вместе с NetCat.

Регистрация

Созданный раздел «Регистрация» содержит системную таблицу «Пользователи»; при этом права на добавление в этот раздел установлены «для всех».

По умолчанию таблица «Пользователи» содержит дополнительные поля: «Email», «Имя пользователя», «Аватар», «Подпись на форуме». Для расширения анкеты пользователя вы можете добавить любое поле в таблицу «Пользователи» в режиме администрирования в разделе «Разработка – Системные таблицы».

После успешной регистрации пользователь может получать просьбу о подтверждении и активации аккаунта (измените настройки модуля: «Настройки» — «Общие настройки» — «Активация», включите опцию «Требовать подтверждение через почту»). Текст письма с подтверждением регистрации, а также любые действия после регистрации можно контролировать в шаблоне действий «добавление» в системной таблице «Пользователи».

После подтверждения регистрации аккаунт пользователя может быть активирован не сразу, а после проверки администратором (измените настройки модуля: «Настройки» — «Общие настройки» — «Активация», включите опцию «Премодерация администратором»). Также можно разрешить пользователям авторизоваться автоматически после подтверждения по почте. Вы можете отправлять пользователю дополнительное уведомление о включении аккаунта при помощи шаблона действий «изменение» («Действие после включения/выключения») в таблице «Пользователи».

Форма регистрации

По умолчанию форма регистрации состоит из полей для ввода логина, пароля, подтверждения пароля и полей, которые были указаны в настройке модуля для заполнения в форме регистрации.

В настройках модуля можно включить возможность проверки доступности логина без перезагрузки страницы, проверки надежности пароля, проверки совпадения паролей.

Практически все HTML-элементы формы регистрации, сгенерированной автоматически, содержат id и классы, чтобы можно было управлять их внешним видом через CSS. Вы можете переопределить форму регистрации через системную таблицу «Пользователи» — «Добавление», но имейте в виду, что java-скрипты, используемые для проверки логина и пароля, привязываются к id и классам, которые были заданы изначально.

Авторизация и завершение сеанса работы пользователя

После того, как аккаунт пользователя активирован, у него появляется возможность авторизоваться на сайте.

Существует несколько способов авторизации:

- обычным способом (с помощью логина и пароля);
- по хэш-коду (авторизация без ввода логина и пароля, по определенной ссылке)
- по usb-токену
- через внешние сервисы (В Контакте, Facebook, Twitter, OpenID)

Подробнее о каждом варианте будет рассказано в соответствующей главе. Рассмотрим авторизацию по логину и паролю.

Авторизация как для администраторов сайта, так и для всех остальных пользователей может быть трех типов – http, cookie и session. Способ авторизации настраивается в файле настроек vars.inc.php в переменной \$AUTHORIZE_BY.

Авторизация типа «http» выглядит как всплывающее окно (поверх окна браузера) с предложением ввести логин и пароль. Используя данный тип авторизации, система может сохранять информацию об авторизованном пользователе только в разделах, закрытых для общего просмотра.

В случае «cookie»-авторизации, информация о пользователе хранится в течение суток или до принудительного завершения сеанса пользователем.

Возможно изменить время хранения «cookie» в файле настроек vars.inc.php. Время задается в переменной \$ADMIN_AUTHTIME, по умолчанию равной 86400 секунд, т.е. 24 часам.

«Session» - это 32 разрядное хэш число, передаваемое в ссылке. Используйте данный тип, если ваш компьютер не может принимать cookie и по какой-либо причине не работает с «http».

Если на страницах сайта необходимо установить форму авторизации, то рекомендуется использовать «cookie» или «session»-авторизацию, так как в этом случае, если пользователь авторизован, на всех страницах сайта вы сможете отобразить, к примеру, персональное приветствие для авторизованного пользователя вместо формы авторизации.

Для вывода формы авторизации в макете есть методы nc_auth::auth_form и nc_auth::auth_link. Первый метод выводит форму авторизации, второй выводит ссылки «Регистрация» и «Авторизация», причем авторизация в этом случае будет идти через всплывающее окно. В любом случае вы можете самостоятельно написать форму авторизации, но она должна содержать следующие поля:

```
<form action='/netcat/modules/auth/' method='post'>
  <input type='text' name='AUTH_USER' /> # логин
  <input type='password' name='AUTH_PW' /> # пароль
  <input type='hidden' name='AuthPhase' value='1' />
  #куда будет перемещен пользователь после авторизации ( в данном
  случае — на текущую страницу )
  <input type='hidden' name='REQUESTED_FROM'
value='$REQUEST_URI' />
  <input type='hidden' name='REQUESTED_BY'
value='$REQUEST_METHOD' />
  <input type='hidden' name='catalogue' value='$catalogue' />
  <input type='hidden' name='sub' value='$sub' />
  <input type='hidden' name='cc' value='$cc' />
  <input type='submit' value='Авторизоваться' />
</form>
```

При необходимости вы можете добавлять другие поля в эту форму (например, если вам нужно передать значения \$template, \$isNaked или др.).

Завершение сеанса может быть представлено как в виде формы, так и в виде ссылки. В виде формы это выглядит так:

```
<form action='/netcat/modules/auth/' method='post'>
```

```
<input type='hidden' name='logout' value='1' />
<input type='hidden' name='REQUESTED_FROM'
value='$REQUEST_URI' />
<input type='hidden' name='REQUESTED_BY'
value='$REQUEST_METHOD' />
<input type='submit' value='Завершить сеанс' />
</form>
```

В виде ссылки так:

```
<a href='/netcat/modules/auth/?logout=1
&REQUESTED_FROM=$REQUEST_URI
&REQUESTED_BY=$REQUEST_METHOD'>
  Завершить сеанс
</a>
```

Изменение регистрационных данных

Каждый авторизованный пользователь имеет доступ к изменению собственной анкеты и пароля. Для этого создается специальный раздел «Изменение данных», причем номер этого раздела указан в настройках модуля. Изменение данных производится аналогично изменению информации о любом объекте в системе. При необходимости вы можете настроить шаблоны действий «условия изменения» и «действие после изменения» для системной таблицы «Пользователи». Вы можете разрешить пользователям менять логин после регистрации в настройках модуля («Настройки» — «Регистрация и авторизация» — «Основные настройки», опция «Разрешить менять логин после регистрации»). Настройка вывода формы изменения профиля происходит через системную таблицу «Пользователи» — «Изменение».

Изменение пароля

Раздел «Изменение пароля» ссылается на скрипт, который позволяет авторизованному пользователю изменить свой пароль. Скрипт имеет адрес «/netcat/modules/auth/password_change.php». Адрес этого скрипта должен быть прописан во внешней ссылке раздела, где пользователь изменяет свой пароль (тогда при заходе на скрипт /netcat/modules/auth/password_change.php будет загружаться именно этот раздел).

Внешний вид формы изменения пароля задается в настройках модуля («Настройки» — «Шаблоны вывода» — «Изменение пароля»). Там же можно задать текст, который выведется при успешном изменении пароля и при ошибке (например, пароль слишком короткий или пароли не совпадают). Сам текст ошибки содержится в переменной `$warnText`. Также в блоке о выводе ошибки доступна переменная `$nc_error_num` — номер ошибки, которая принимает значения:

- 1 — если пароли не совпадают
- 2 — если пароль пустой
- 3 — если пароль слишком короткий

Восстановление пароля

Модуль также содержит функцию восстановления пароля. Чтобы активизировать данную возможность, вам необходимо создать раздел, в котором будет ссылка (в поле «внешняя ссылка») на скрипт `/netcat/modules/auth/password_recovery.php`

Внешний вид формы восстановления пароля задается в настройках модуля («Настройки» — «Шаблон вывода» — «Восстановление пароля»). На этой же странице настроек задается текст, который будет показан пользователю, после того, как он получит письмо со ссылкой на восстановление пароля и блок вывода ошибок. Как и в блоке вывода ошибок, при изменении пароля, в этом блоке доступны две переменные:

`$warnText` — текст ошибки

`$nc_error_num` — номер ошибки:

- 1 — поля не заполнены
- 2 — пользователь не найден

Отображение списка пользователей на сайте

Для отображения списка пользователей в каком-либо разделе вам необходимо подключить к этому разделу таблицу «Пользователи». При создании компонента раздела вы увидите таблицу «Пользователи» в конце списка. Настройка отображения списка пользователей, а также информация о них на отдельной странице аналогична настройке компонентов в разделе «Разработка — Системные таблицы — Пользователи».

В стандартной поставке системная таблица «Пользователи» выводит пользователей посредством таблицы из 6 колонок: аватар, имя пользователя

NetCat 4.5 Руководство по модулям

с ссылкой на профайл, приватное сообщение (если эта опция разрешена в настройках модуля), друг/враг (если эта опция включена в настройках модуля), дата регистрации.

Перед списком пользователей есть форма поиска (она задается в системной таблице «Пользователи» на вкладке «Поиск») по двум критериям: имя или логин + «только онлайн», выборка пользователей по поисковым критериям происходит в системных настройках, при желании их можно модифицировать.

Отображение пользователей присутствующих на сайте

Для отображения пользователей, присутствующих на сайте, необходимо использовать функцию:

```
nc_auth_users_online($template = NULL, $select_fields = NULL)
```

`$template` - шаблон вывода списка пользователей или режим вывода.
`$template` может содержать различные значения:

nc_auth_users_online()
Вернет строку со списком пользователей (шаблоном по умолчанию)
nc_auth_users_online (ARRAY_A)
Вернет массив с идентификаторами пользователей К примеру: Array ([0] => 1, [2] => 3, [3] => 7)
nc_auth_users_online (ARRAY_N) nc_auth_users_online (ARRAY_N, 'u.Email, s.SessionStart')
Вернет массив со следующими значениями: ●User_ID (идентификатор пользователя) ●Name (имя пользователя) ●Url (ссылка на профиль пользователя) При этом в переменной <code>\$select_fields</code> можно указать альтернативные поля из таблиц User (u) и/или Session (s), которые будут выбраны вместо поля с именем пользователя. К примеру:

<pre>\$select_fields = 'u.Email, s.SessionStart';</pre> вернет массив со следующими значениями: ●User_ID (идентификатор пользователя) ●Email (адрес эл. почты пользователя) ●SessionStart (время в формате unix time с момента авторизации) ●Url (ссылка на профиль пользователя)
nc_auth_users_online(\$my_template) Вернет строку со списком пользователей (заданные шаблон). Сам шаблон должен быть задан в виде массива: <pre>\$my_template['prefix'] = ""; \$my_template['suffix'] = ""; \$my_template['divider'] = " "; \$my_template['link'] = "%NAME";</pre> при этом только параметр \$my_template['link'] является обязательным. В самом шаблоне доступны следующие переменные: %NAME — имя пользователя %URL — адрес профиля пользователя %ONLINE — количество пользователей, присутствующих на сайте %REGISTERED — количество зарегистрированных пользователей, присутствующих на сайте %GUESTS — количество гостей nc_auth_users_online("Сейчас на сайте: %ONLINE пользователей (зарегистрированных: %REGISTERED, гостей: %GUESTS) ") Вернет строку следующего вида: Сейчас на сайте: 12 пользователей (зарегистрированных: 7, гостей: 5)

Личные сообщения

Для использования личных сообщений необходимо добавить к какому-либо разделу компонент: «Пользователи: Личные сообщения». В настройках модуля можно запретить использование личных сообщений.

Чтобы вывести количество новых сообщений для текущего пользователя необходимо использовать функцию:

```
nc_auth_messages_new()
```

Функция имеет дополнительные необязательные параметры:

```
nc_auth_messages_new($User_ID = 0, $Sub_Class_ID = 0)
```

`$User_ID` — идентификатор пользователя

`$Sub_Class_ID` — номер (идентификатор) компонента личных сообщений в разделе. Необходимо указывать, если на сайте присутствуют 2 или более раздела с личными сообщениями.

Ссылку на раздел с личными сообщениями можно получить при вызове функции:

```
nc_auth_messages_url()
```

при этом нужно вызвать функцию с идентификатором пользователя:

```
nc_auth_messages_url(12)
```

функция вернет ссылку на форму отправки личного сообщения пользователю с таким идентификатором.

Нотификация

В настройках модуля можно включить оповещение пользователя по email о новом личном сообщении. Письмо редактируется и отправляется в «Действии после добавления».

Авторизация по хэшу

Модуль позволяет пользователю авторизоваться без ввода логина и пароля, зайдя на сайт по определенной ссылке. Ссылка может выглядеть так:

http://example.com/feedback/?auth_hash=53228d4d1ce43f72252f9b7717bd83c910

Естественно, хэш-ключ будет другой. Хэш передается через параметр `auth_hash`, и, если есть пользователь, которому соответствует этот код, он станет авторизованным.

В настройках модуля можно включить/выключить возможность авторизации по хэшу, задать «время жизни» хэш-кода в часах после создания.

Параметры, определяющие время жизни хэша и возможности многократной авторизации, можно переопределить при непосредственном создании хэша, речь об этом пойдет ниже.

Для некоторых разделов можно запретить авторизацию по хэшу. Допустим, раздел с номером 5 и ключевым словом `bank` служит для управления лицевым счетом пользователя и требуется запретить в нем авторизоваться по хэшу. В этом случае в настройках модуля («Настройки» — «Системные» — «Хэш-авторизация») нужно через запятую перечислить номера всех разделов, где авторизация по хэшу запрещена. В данном случае это просто 5.

Пользователь заходит на страницу «О компании» по адресу [example.com/about/?](http://example.com/about/?auth_hash=34523458s4dfcdg4df370452f92771223bd8ac910)

[auth_hash=34523458s4dfcdg4df370452f92771223bd8ac910](http://example.com/about/?auth_hash=34523458s4dfcdg4df370452f92771223bd8ac910)

(предполагается, что существует пользователь с таким хэш кодом), и система авторизует его там, но при переходе на страницу управления счетом example.com/bank/ пользователь перестает быть авторизованным (для того, чтобы заново авторизоваться, ему нужно будет снова ввести логин и пароль). Причем если пользователь перейдет на другую страницу (где авторизация по хэшу разрешена), он снова станет авторизованным.

Для создания и удаления хэш-кодов существуют методы. Чтобы ими воспользоваться, нужно сначала получить объект класса **`nc_auth`**.

По умолчанию объект этого класса уже существует в глобальной области видимости и имеет имя `$nc_auth`. Для доступа к этому объекту из функций необходимо либо сделать ее глобальной:

```
global $nc_auth
```

либо вызвать статический метод для получения объекта:

```
$nc_auth = nc_auth::get_object();
```

В дальнейшем будем предполагать, что объект `$nc_auth` доступен. Функции для работы с хэш-авторизацией доступны через объект `hash`, который является членом объекта `$nc_auth`.

Для создания хэша существует функция **`create_auth_hash`**

Ее вызов:

```
str $nc_auth->hash->create_auth_hash( $user_id, $delete_hash = null, $expire = null )
```

параметры:

user_id (целое число) — номер пользователя, для которого создается хэш;
delete_hash (булево значение) — удалять ли хэш после первого захода (если параметр не передан, то берется значение из настроек модуля). Если параметр равен **true**, то повторная авторизация по этому хэшу будет невозможна;
expire (целое число) — время жизни хэша в часах (если параметр не передан, то берется значение из настроек модуля).

Функция возвращает строку с хэш-кодом.

Пример использования:

Необходимо создать хэш код для пользователя с идентификатором 3:

```
$nc_auth = nc_auth::get_object();  
$hash = $nc_auth->hash->create_auth_hash (3, 1, 24*3);
```

переменная `$hash` будет содержать хэш-код, и если передать его через параметр `auth_hash`, пользователь будет авторизован с номером 3, причем не важно, в каком разделе будет происходить авторизация. Авторизоваться по этому хэшу можно будет только один раз в течение трех дней.

Если отправить ссылку с этим хэшем пользователю:

```
$mailer = new CMIMEMail();  
$mailer->mailbody( «Пройдите по ссылке  
http://example.com/somesub/?auth_hash=».$hash);  
$mailer->send(«e-mail пользователя 3», «Admin@example.com»,  
«Admin@example.com», «Новое сообщение», «Adim»);
```

(подробнее о отправке писем смотрите в «Руководстве разработчика»)

пользователю придет письмо следующего вида:

«Пройдите по ссылке
`http://example.com/somesub/?auth_hash=4bcdce0f65cdf542e29a327e06fc132811347174`»

пройдя по которой, он станет авторизованным пользователем.

Если требуется удалить хэш для пользователя, то нужно вызвать метод **delete_auth_hash**

`$nc_auth->hash->delete_auth_hash( $user_id )`
`user_id` — номер пользователя, хэш которого нужно удалить.

Пример:

```
$nc_auth->hash->delete_auth_hash(3);
```

удалит хэш для пользователя с идентификатором 3.

Чтобы узнать хэш пользователя, можно воспользоваться методом **get_hash_by_user**

`$nc_auth->hash->get_hash_by_user( $user_id )`
`user_id` — номер пользователя, для которого нужно узнать хэш.

Пример:

```
$hash = $nc_auth->hash->get_hash_by_user(3);
```

В переменной `$hash` будет содержаться хэш для пользователя с идентификатором 3.

В массиве `$current_user`, хранящем информацию о текущем авторизованном пользователе, содержится также информация о варианте авторизации, она доступна через ключ `AuthVariant`. Если `$current_user['AuthVariant']` константе `NC_AUTHTYPE_HASH`, то пользователь авторизовался с помощью хэша. Подробнее о массиве `$current_user` рассказано ниже.

Сам хэш код хранится в таблице `User` в поле `Auth_Hash`.

Примечания:

Функционал работает при типе авторизации «cookie» или «session». Непосредственно авторизоваться по хэшу в системе управления нельзя.

Авторизация через внешние сервисы

Модуль позволяет пользователям авторизоваться через:

- В Контакте
- Facebook

- Twitter
- OpenID (Mail.ru, Yandex.ru, Google.com, Myopenid.com и т.д.)

Пользователь, единожды авторизовавшись через один из этих сервисов, попадает в базу пользователей вашего сайта, в последствии он может задать себе логин/пароль или изменить профайл (например, сменить аватар).

Настройка такого типа авторизации происходит через настройки модуля «Личный кабинет» — «Регистрация и Авторизация» — «Через внешние сервисы». Для каждого сервиса задается:

- группы, куда попадет пользователь после авторизации;
- действие после первой авторизации;
- соответствие полей.

Действие после первой авторизации содержит php-код, который выполнится после того, как пользователь будет авторизован и занесен в базу пользователей сайта. В этом блоке доступны переменные и массивы:

`$user_id` — номер пользователя в системе NetCat;

`$ex_user_id` — идентификатор пользователя в сервисе, по которому идет авторизация (например, в В Контакте — это номер, в OpenID – логин);

`$userinfo` — массив с данными о пользователе, который передал сервис;

`$db` — объект для работы с БД;

`$nc_core` — системный объект класса `nc_Core`.

Сервисы могут передавать некоторую информацию о пользователе (например, ник, имя, фамилия, фотография, email). Эту информацию можно записать в профайл авторизованного пользователя, для этого воспользуйтесь блоком «Соответствие полей».

Примечание. Данные о пользователе от сервиса могут и не передаваться, все зависит от самого сервиса, плюс ко всему пользователь может запретить передавать информацию о нем стороннему сайту.

Авторизация через В Контакте

Для того, чтобы пользователи могли авторизоваться на вашем сайте через В Контакте, вам необходимо сначала зарегистрировать приложение на сайте `vkontakte.ru`. На момент написания документации ссылка на добавление приложения была такая:

<http://vkontakte.ru/apps.php?act=add&site=1>

После регистрации приложения вы получите ID приложения и защищенный ключ:

Редактирование Реклама Переводы

ID приложения: **2012310**

Защищенный ключ:

[Редактировать](#)

Настройки Open API

Адрес сайта:

Базовый домен:

[Сохранить](#)

Удаление приложения

Если Вы удалите это приложение, Вы уже не сможете его восстановить.

[Удалить приложение](#)

Информация
Настройки
Администраторы

Скопируйте эти параметры и вставьте в настройки модуля в поля ID приложения и защищенный ключ в области «В Контакте»:

Личный кабинет

Информация **Регистрация и авторизация** Настройки

По логину и паролю **Через внешние сервисы**

▼ **ВКонтакте:**

☒ Включить авторизацию через vkontakte.ru

ID приложения:

Защищенный ключ:

Группы, куда попадёт пользователь:

- ☐ Администраторы
- ☐ Внешние пользователи
- ☐ Псевдопользователи
- ☒ Авторизованные через внешние сервисы

После чего на сайте можно будет авторизоваться через vkontakte.ru. Если для вывода формы авторизации вы используете метод `auth_form`, то он автоматически покажет кнопку «Войти через В Контакте»:



При использовании своей формы авторизации можете воспользоваться следующим кодом для показа кнопки «Авторизоваться через В Контакте»:

```
<div id='login_button' class='vk_login_button'
  onclick='nc_vk_login("$.nc_auth_vk->is_member().")'></div>
<script src='http://vkontakte.ru/js/api/openapi.js' type='text/javascript'></script>
<script type='text/javascript'>
  function nc_vk_login ( is_member ) {
    var is_mem = is_member || 0;
    if ( !is_mem ) VK.Auth.login( nc_vk_login );
    else
      location.href="$.nc_core->SUB_FOLDER.nc_core-
>HTTP_ROOT_PATH."modules/auth/?
nc_vk=1&REQUESTED_FROM=".htmlspecialchars($REQUEST_URI,ENT_Q
UOTES ).";
  }
  VK.init({ apiId: "$.nc_auth_vk->get_app_id()." });
  VK.UI.button('login_button');
</script>
```

Авторизация через Facebook

По аналогии с авторизацией через В Контакте, вам надо предварительно создать новое приложение на сайте facebook.com. На момент написания документации это можно было сделать по ссылке <http://www.facebook.com/#!/developers/createapp.php>. После создания приложения вы получите номер приложения (Application ID) и его секретный ключ (Application Secret).



netcat.ru

Directory Status: Not Submitted

Once you have completed your app, you may [submit it](#) to the App Directory.

Количество активных
пользователей за месяц

1

Следующим пользователям это
нравится

0

Всего пользователей:

1

ID Приложения

154800771230966

Ключ API

3af221c49e8113e86160764f1962ee84

App Secret

edc6e1d73060d0f0e0ae4a47b77d26a6

Site URL

<http://netcat.ru/>

Site Domain

netcat.ru

[Изменить настройки](#)

[Application Profile Page](#)

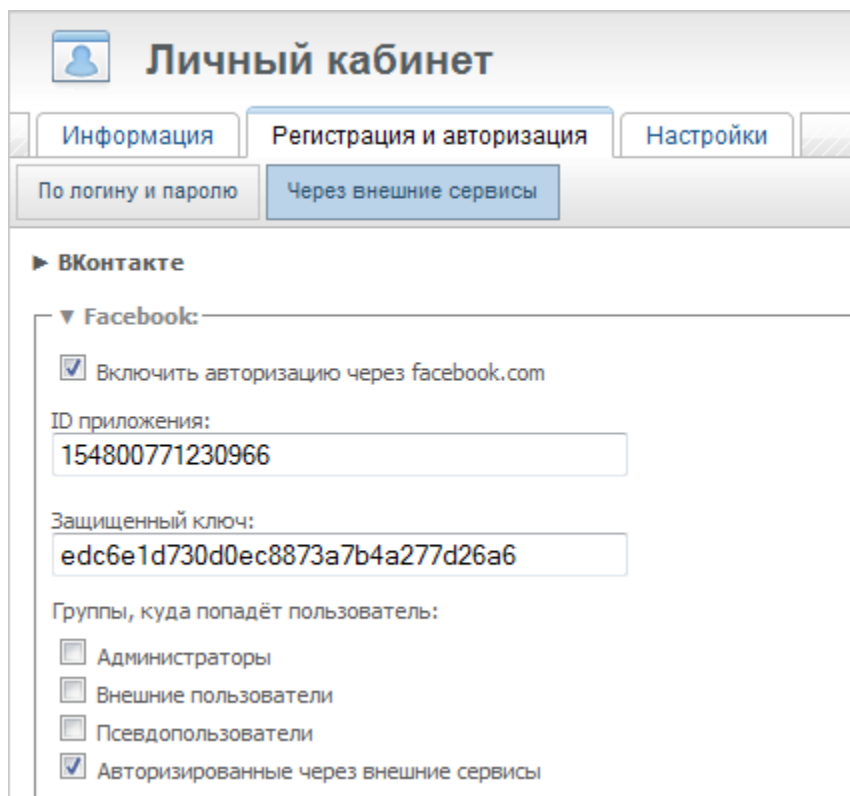
[Исследования](#)

[Переводы](#)

[Разместить рекламу](#)

[Reset App Secret](#)

Эти параметры надо занести в настройки модуля:



После чего на сайте можно будет авторизоваться через facebook.com. Если для вывода формы авторизации вы используете метод `auth_form`, то он автоматически покажет кнопку Facebook «Войти»:



При использовании своей формы авторизации можете воспользоваться следующим кодом для показа кнопки авторизации:

```
<p class='fb'>
  <fb:login-button perms=\"email\" \".( $nc_auth_fb->get_token() ?
  \"onclick='location.href=\"\".$nc_core->SUB_FOLDER.\"/netcat/modules/auth/?
  nc_fb=1&REQUESTED_FROM=\".htmlspecialchars($REQUEST_URI,ENT_Q
  UOTES ).\"\" : \"\").\" ></fb:login-button>
</p>
<div id='fb-root' ></div>
<script type='text/javascript'
src='http://connect.facebook.net/ru_RU/all.js'></script>
<script type='text/javascript'>
  FB.init({appId: \"\".$nc_auth_fb->get_app_id().\", cookie:true, xfbml: true});
  FB.Event.subscribe('auth.login', function(response) { location.href = \"\".
  $nc_core->SUB_FOLDER.$nc_core->HTTP_ROOT_PATH.\"modules/auth/?
```

```
nc_fb=1&REQUESTED_FROM=".htmlspecialchars($REQUEST_URI,ENT_QUOTES)."; });  
</script>
```

Авторизация через Twitter

Для авторизации через твиттер вам необходимо создать новое приложение по ссылке (действительна на момент написания документации):
<http://twitter.com/apps/new>

После успешной регистрации приложения, вы получите consumer key и consumer secret:

Application Details



netcat.ru by

Приложение для авторизации через twitter

created by **denis varin** – read-only access by default

Edit Application Settings

Reset Consumer Key/Secret

Consumer key

wHAc76dOS5bmwXGoN0hMg

Consumer secret

NGgUkg9qxZdj9d1yx6FEiju4O1FKzZwoGrlxa44ezcU

Request token URL

http://twitter.com/oauth/request_token

Access token URL

http://twitter.com/oauth/access_token

Authorize URL

<http://twitter.com/oauth/authorize> * We support hmao-sha1 signatures. We do not support the plaintext signature method.

Скопируйте эти параметры и вставьте в настройки модуля:


Личный кабинет

Информация
Регистрация и авторизация
Настройки

По логину и паролю
Через внешние сервисы

▼ Twitter:

☒ Включить авторизацию через twitter.com

ID приложения:

Защищенный ключ:

Группы, куда попадѣт пользователь:

☐ Администраторы
☐ Внешние пользователи
☐ Псевдопользователи
☒ Авторизированные через внешние сервисы

Действие после первой авторизации пользователя:

Теперь чтобы пользователь авторизовался через твиттер, ему надо пройти по ссылке

/netcat/modules/auth/?nc_twitter=1

То есть форму авторизации будет выглядеть вот так:

```
<a href='/netcat/modules/auth/?nc_twitter=1'>
  <img src='Путь до значка твиттера' alt='Авторизоваться через твиттер' />
</a>
```

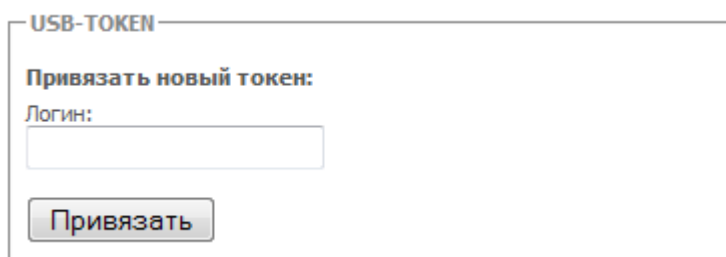
Авторизация через USB-токены

В NetCat реализована возможность авторизации через webtoken (<http://www.webtoken.ru/>). Идея заключается в том, что usb-токен связывается с пользователем NetCat (например, с администратором сайта), и в дальнейшем токен используется для авторизации, причем систему можно настроить таким образом, что администратор сайта сможет зайти в админ-NetCat 4.5 Руководство по модулям

панель только имея токен, что значительно более безопасно. Подробнее о самих web-токенах можно узнать на приведенном выше сайте. Для авторизации через токен нужно установить плагин к браузеру, скачать его можно на странице <http://www.webtoken.ru/download>

Привязка пользователя к токenu

Привязка пользователя к токenu осуществляется через систему администрирования. Вставьте usb-токен в компьютер, в списке пользователей («Пользователи» — «Пользователи») выберите нужного пользователя. В самом конце страницы редактирования пользователя будет блок «USB-TOKEN»:



USB-TOKEN

Привязать новый токен:

Логин:

Привязать

Введите логин, который будет записан на токен (он необязательно должен совпадать с логином пользователя), например, admin, и нажмите «Привязать», далее в диалоговом окне введите PIN-код токена, после чего информация о пользователе сохранится, и пользователь будет привязан к токenu. Теперь данный пользователь может авторизоваться с помощью токена.

Внимание. Если вы несколько раз введете неправильный PIN-код, то токен заблокируется.

В том же блоке можно увидеть, к каким пользователям привязан токен и при желании «отсоединить» этого пользователя от токена.

Внимание. Если вы несколько раз введете неправильный PIN-код, то токен заблокируется.

Форма авторизации по токену на сайте

Для авторизации по токену на сайте вы можете использовать следующую форму:

```
<object id='nc_token_plugin' type='application/x-rutoken' width='0'
height='0'></object>
<!-- в этот элемент будут писать тексты ошибок -->
<div id='nc_auth_token_info'></div>
<form id='nc_token_form'>
  <!--этот контейнер по умолчанию скрыт, показывается, если токен
вставлен-->
  <div id='nc_auth_token_forn' style='display:none;'>
    <select name='nc_token_login' id='nc_token_login'></select>
    <input type='hidden' value="" id='nc_token_signature'
name='nc_token_signature' />
    <input class='auth_submit' onclick='nc_token_sign(); return false;'
type='button' value='Войти по токену' />
  </div>
</form>

<script type='text/javascript'>
  // инициализируем объект для работы с токенами
  nc_token_obj = new nc_auth_token ( {'randnum' : ''.$nc_auth_token-
>generate_rand().''});
  // если токен вставлен, то покажем форму с выбором логина
  if ( nc_token_obj.load() ) $('#nc_auth_token_form').show();
// авторизация
  function nc_token_sign ( ) {
    $('#nc_auth_token_info').hide();
    // коды ошибок
    err_text = {
1: 'Плагин не установлен',
2: 'Токен не вставлен',
```

```

3: 'Пин-код введен неверно',
4: 'Произошла ошибка'
};
    if ( (err_num = nc_token_obj.sign()) ) {
        // если произошла ошибка — покажем ее
        $('#nc_auth_token_info').html(err_text[err_num]);
        $('#nc_auth_token_info').show();
    }
}
</script>

```

Друзья-Враги

Описание

Модуль позволяет пользователям выбирать себе друзей или врагов. Таким образом, можно ограничить просмотр объектов пользователя другими пользователями (например, только для друзей или всем, кроме врагов). Фактически система представляет собой API.

С этим модулем связан компонент «Пользователи: Друзья». Этот компонент выводит на страницу всех пользователей (компонент устроен так, что выборка записей идет не из таблицы MessageXX, а из таблицы User, это сделано через \$ignore_all, подробнее — см. «Руководство разработчика», глава «Системные настройки компонента») и форму для сохранения результата (при наличии модуля «Кэширование» запретите кэширование для раздела, в котором используется этот компонент). Напротив каждого пользователя есть переключатели «Друг», «Нейтраль», «Враг». Все переключатели и кнопка «ОК» находятся в одной форме, действие формы идет на скрипт модуля. Сам компонент «Пользователи: Друзья» вы можете менять, как угодно (например, заменив радиокнопки на выпадающий список или ограничив в системных настройках компонента выборку пользователей), но формат данных, попадающих на скрипт, должен иметь определенный вид. Каким образом данные попадают на скрипт, (через элементы формы и скрытые поля, через get-параметры) не важно. Сам скрипт находится по адресу `/netcat/modules/auth/add_relation.php`

Параметры, передаваемые на скрипты, должны выглядеть следующим образом:

`user_X=y`

где X — номер пользователя

NetCat 4.5 Руководство по модулям

у — тип отношения (1 — сделать другом, 2 — сделать врагом, -1 — удалить отношение)

redirect_url=адрес_страницы

этот параметр определяет, куда перебросить пользователя после сохранения результата выбора друзей/врагов.

В компоненте «Пользователи: Друзья» `redirect_url` задается в скрытом поле:

```
<input type='hidden' value='".$_SERVER['REQUEST_URI']."'
name='redirect_url'>
```

Таким образом после сохранения система перебросит пользователя на эту же страницу.

Как было сказано выше: не важно, как именно данные передаются на скрипт, их можно передать через GET-параметры, то есть пользователь проходит по определенной ссылке, а система добавляет к нему, например, друга.

Ссылка может иметь вид:

http://адрес_сайта/netcat/modules/auth/add_relation.php?user_3=1

Пользователь, зашедший на такую ссылку, добавит к себе в друзья пользователя с номером 3.

Следующая ссылка:

http://адрес_сайта/netcat/modules/auth/add_relation.php?user_35=2&user_2=1&user_5=-1&redirect_url=http://адрес_сайта/profile/

добавит пользователя 35 во враги по отношению к текущему пользователю, добавит в друзья пользователя 2, уберет из друзей/врагов пользователя 5 и перебросит на страницу http://адрес_сайта/profile/.

Если возникает задача вывести в списке пользователей ссылку на «добавление в друзья» (по аналогии с «сообщением»), то решить ее можно следующим образом: в системной таблице «Пользователи», в «Объекте в списке» прописать:

```
<a href='/netcat/modules/auth/add_relation.php?user_".
$f_RowID."=1&redirect_url=".$_SERVER['REQUEST_URI']."'>в друзья</a>
```

Функции модуля и константы

Константы:

NC_AUTH_FRIEND = 1 — друг

NC_AUTH_BANNED = 2 — враг

NC_AUTHTYPE_LOGIN = 1 — авторизация по логину и паролю

NC_AUTHTYPE_HASH = 2 — авторизация по хэшу

NC_AUTHTYPE_EX = 4 — авторизация через внешние сервисы

NC_AUTHTYPE_TOKEN = 8 — авторизация через токен

NC_AUTH_LOGIN_OK = 0 — логин корректен

NC_AUTH_LOGIN_INCORRECT = 1 — логин некорректный

NC_AUTH_LOGIN_EXISTS = 2 — такой логин уже есть

Функции модуля

nc_auth_add_relation (\$type, \$Related_ID, \$User_ID = 0)

Добавляет отношение.

\$type — тип: друг или враг.

\$Related_ID — тот, к кому применяется отношение.

\$User_ID — номер пользователя, по умолчанию текущий.

Примеры:

nc_auth_add_relation(NC_AUTH_FRIEND, 5)

Пользователь с номером 5 становится другом текущего пользователя.

nc_auth_add_relation(NC_AUTH_BANNED, 1, 37)

Пользователь с номером 1 становится врагом пользователя 37.

nc_auth_add_friend (\$Related_ID, \$User_ID = 0)

Пользователь Related_ID становится другом пользователя User_ID.

nc_auth_add_bann (\$Related_ID, \$User_ID = 0)

Пользователь Related_ID становится врагом пользователя User_ID.

nc_auth_get_all_relation (\$User_ID = 0)

Функция возвращает все отношения пользователя (по умолчанию — текущего) в виде массива: НомерПользователя => ТипОтношения.

nc_auth_get_all_friend (\$User_ID = 0)

Возвращает в виде массива всех пользователей, являющихся друзьями пользователя \$User_ID.

nc_auth_get_all_banned (\$User_ID = 0)

Возвращает в виде массива всех пользователей, являющихся врагами пользователя \$User_ID.

nc_auth_is_friend (\$Related_ID, \$User_ID = 0)

Функция позволяет узнать, является ли пользователь \$Related_ID другом User_ID.

nc_auth_is_banned (\$Related_ID, \$User_ID = 0)

Функция позволяет узнать, является ли пользователь \$Related_ID врагом User_ID.

nc_auth_get_relation (\$Related_ID, \$User_ID = 0)

Функция позволяет узнать тип отношения между пользователями. Если Related_ID является другом User_ID, то функция вернет NC_AUTH_FRIEND, если врагом — NC_AUTH_BANNED, если никем — то 0.

nc_auth_is_mutual (\$Related_ID, \$User_ID = 0)

Функция вернет NC_AUTH_FRIEND, если пользователи взаимные друзья, NC_AUTH_BANNED если пользователи взаимные враги и 0 в остальных случаях.

nc_auth_delete_relation (\$Related_ID, \$User_ID = 0)

Функция удаляет отношение между пользователями Related_ID и User_ID.

Пример использования

Покажем, как можно использовать этот функционал для ограничения вывода контента определенным пользователям. Допустим, на нашем сайте пользователи могут сами добавлять объекты в некоторые разделы (новости, личные блоги) и эти сообщения необходимо показывать только друзьям. Тогда в соответствующем компоненте в «Объекте в списке» нужно применить конструкцию такого плана:

```
".( $AUTH_USER_ID &&( $AUTH_USER_ID == $f_UserID ||  
nc_auth_is_friend($AUTH_USER_ID, $f_UserID) )? "вывод сообщения" : "")."
```

`$AUTH_USER_ID` – содержит номер текущего авторизованного пользователя.

`$f_UserID` – номер пользователя/автора сообщения.

Личный счет

Модуль предоставляет средства для ведения личного счета пользователя. Личный счет можно включить/выключить в настройках модуля, там же можно задать единицу измерения счета (например, рубли или у.е.), а так же количество единиц, которые будет зачислены пользователю при регистрации.

Баланс пользователя реализован через стандартные поля системной таблицы «Пользователи», то есть вы можете оперировать им как обычным полем. Данная часть модуля предоставляет API для работы с балансом (начисление и списание средств, описание функций будет приведено ниже), а также компонент («Пользователи: Личный счет») для хранения истории изменения счета, в котором каждый объект предоставляет собой транзакцию:

Ваш текущий баланс: 125

Тип операции	Дата и время	Сумма	Примечание
пополнение	11.02.2011 12:30:00	100	Кристаллический хребет: предпосылки и развитие
снятие	12.02.2011 12:41:00	75	Полого-холмистый сталактит: основные моменты

Если вы добавите объект в этот компонент (при добавлении надо указать пользователя, тип операции: пополнение или снятие, сумму и примечание), то личный счет этого пользователя автоматически изменится. Также существуют программные средства управления счетами:

`$nc_auth->pa_add ( $sum, $user_id = 0, $desc = '', $dont_log = 0 )`

Вносит средства в количестве `$sum` пользователю с номером `$user_id`.

Параметры:

- `$sum` – сумма
- `$user_id` – номер пользователя (по умолчанию — текущий)
- `$desc` – описание транзакции (обязательно)
- `$dont_log` – не вносить запись в компонент с личным счетом

Метод возвращает баланс пользователя после проведения операции.

`$nc_auth->pa_deduct ( $sum, $user_id = 0, $desc = '', $dont_log = 0 )`

Списывает средства в количестве `$sum` у пользователя с номером `$user_id`.

Параметры:

- \$sum – сумма
- \$user_id – номер пользователя (по умолчанию — текущий)
- \$desc – описание транзакции (необязательно)
- \$dont_log – не вносить запись в компонент с личным счетом

Метод возвращает баланс пользователя после проведения операции.

Административная часть модуля

Структура этой части сходна со структурой административной части модуля.

Информация

На данной вкладке отображается информация об общем количестве пользователей, о количестве выключенных пользователей и о количестве пользователей, еще не подтвердивших регистрацию.

Регистрация и авторизация

Данная вкладка содержит два раздела:

- По логину и паролю
- Через внешние сервисы

По логину и паролю

Основные настройки:

Основные настройки

- ☐ Запретить самостоятельную регистрацию
- ☐ Запретить самостоятельно восстанавливать пароли
- ☐ Разрешить кириллицу в логинах
- ☒ Разрешить спецсимволы в логинах
- ☐ Разрешить менять логин после регистрации
- ☐ Привязывать пользователя к сайту
- ☒ Авторизировать на поддоменах

Показывать CAPTCHA после попыток авторизации (0 - не показывать)

Минимальная длина пароля символов

«Запретить самостоятельную регистрацию» — при включении этой опции пользователи не смогут регистрироваться через сайт, создать нового пользователя можно будет только через систему администрирования.

«Запретить самостоятельно восстанавливать пароли» — при включении этой опции пользователи не смогут самостоятельно (через почту) восстанавливать утерянные пароли. При попытке восстановить пароль им будет показан текст, заданный в настройках модуля («Настройки» — «Шаблоны вывода» — «Восстановление пароля» — «Текст, выводимый при запрете восстановления пароля»).

«Разрешить кириллицу в логинах» — при включении данной опции пользователи смогут использовать кириллицу в своих логинах.

«Разрешить спецсимволы в логинах» — при включении данной опции пользователи смогут использовать спецсимволы (например, %) в своих логинах.

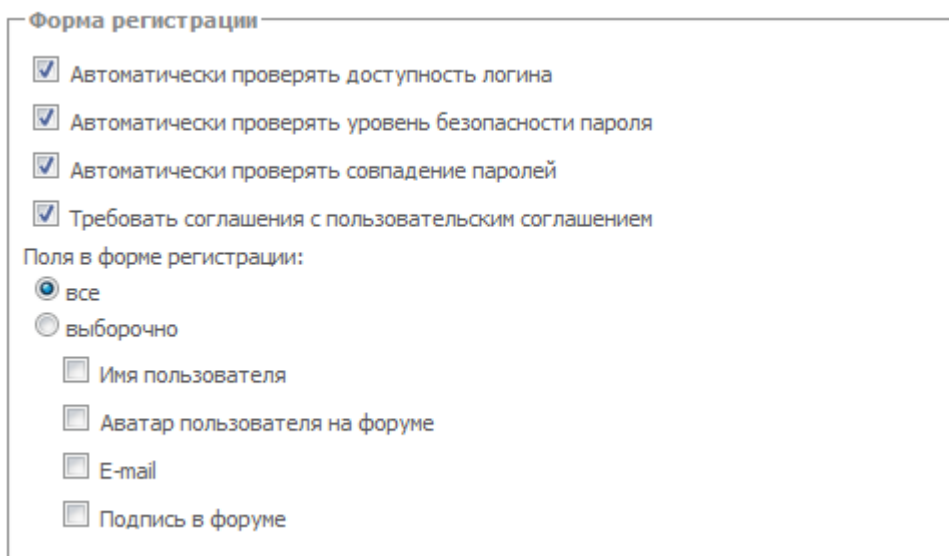
«Разрешить менять логин после регистрации» — при включении данной опции пользователи смогут менять логин, по которому они авторизовались, в своем профайле.

«Привязывать пользователя к сайту» — при включении данной опции пользователи, зарегистрированные на одном из сайтов системы, не смогут авторизоваться на другом сайте этой же системы.

«Авторизовывать на поддоменах» — при включении данной опции пользователь, авторизованный на основном домене сайта (например, netcat.ru), будет авторизован и на всех его поддоменах (например, developer.netcat.ru). Если поддомен представляет с собой отдельный сайт в

системе администрирования, то предыдущая опция «привязывать пользователя к сайту» должна быть выключена.

Форма регистрации:



Форма регистрации

- ☒ Автоматически проверять доступность логина
- ☒ Автоматически проверять уровень безопасности пароля
- ☒ Автоматически проверять совпадение паролей
- ☒ Требовать соглашения с пользовательским соглашением

Поля в форме регистрации:

☒ все

☐ выборочно

- ☐ Имя пользователя
- ☐ Аватар пользователя на форуме
- ☐ E-mail
- ☐ Подпись в форуме

В этом блоке настраивается форма регистрации.

«Автоматически проверять доступность логина» — при регистрации по мере набора логина будет показана информация о доступности этого логина (логин свободен, логин занят, логин содержит запрещенные символы).

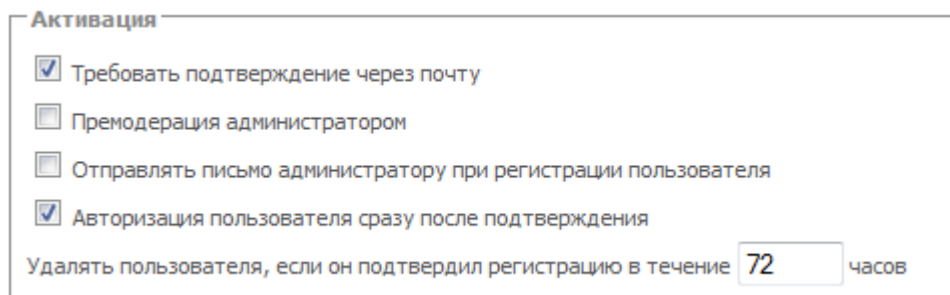
«Автоматически проверять уровень безопасности пароля» — при включении этой опции по мере набора пароля при регистрации будет

показан его уровень надежности (низкий уровень, средний, высокий, очень высокий).

«Автоматически проверять совпадение паролей» — при включении этой опции по мере набора подтверждения пароля будет показано, совпадает ли он с основным паролем.

«Требовать соглашения с пользовательским соглашением» — при регистрации будет необходимо принять условия соглашения.

Активация:



Активация

- ☒ Требовать подтверждение через почту
- ☐ Премодерация администратором
- ☐ Отправлять письмо администратору при регистрации пользователя
- ☒ Авторизация пользователя сразу после подтверждения

Удалять пользователя, если он подтвердил регистрацию в течение часов

«Требовать подтверждение через почту» — после регистрации пользователю придет письмо на почту с ссылкой на подтверждение регистрации.

«Премодерация администратором» — при включении этой опции пользователи после регистрации (и после подтверждения) всегда будут выключенными, пока администратор сайта сам их не включит.

«Отправлять письмо администратору при регистрации пользователя» — при включении этой опции администратору проекта будут приходить оповещения, что новый пользователь авторизован.

И последняя настройка в этом блоке — «время жизни» пользователя, неподтвердившего регистрацию по почте. По истечении этого срока он будет удален. Если поставить 0, то пользователи удаляться не будут.

В последнем блоке можно задать группы, куда попадет пользователь после регистрации:

Группы, куда попадёт пользователь

- ☐ Администраторы
- ☒ Внешние пользователи
- ☐ Псевдопользователи
- ☐ Авторизованные через внешние сервисы

Через внешние сервисы

На данной странице можно настроить авторизацию через внешние сервисы. Подробнее о них было рассказано в главе «Авторизация через внешние сервисы».

Настройки

Данная вкладка содержит 4 раздела:

- общие
- шаблоны вывода
- шаблоны писем
- системные

Общие

В первых двух блоках можно выбрать какими способами возможно авторизоваться на сайте и в систему администрирования:

Способы авторизации на сайте

- ☒ По логину и паролю
- ☒ По хэшу
- ☐ Через внешние сервисы
- ☐ По usb-токену

Способы авторизации в систему администрирования

- ☒ По логину и паролю
- ☐ По хэшу
- ☒ По usb-токену

Следующие два блока — это включение возможности отправки личных сообщений, а также возможность добавлять пользователей в друзья/враги:

Личные сообщения

☒ Разрешить отправлять личные сообщения
☒ Оповещать пользователя по e-mail о новом сообщении

Друзья и враги

☒ Разрешить добавлять пользователей в друзья
☐ Разрешить добавлять пользователей во враги

И в последнем блоке на этой странице настраивается личный счет пользователя

Личный счет

☒ Ввести личный счет
 Единица измерения
 Начислять вновь зарегистрированным единиц

Шаблоны вывода

На данной странице настраиваются:

- форма авторизации на отдельной странице
- форма изменения пароля
- форма восстановления пароля

Шаблоны писем

На данной странице настраиваются шаблоны писем, которые используются для:

- подтверждения регистрации
- восстановления пароля
- оповещения администратора о новом пользователе

В шаблонах писем доступны следующие макропеременные:

%SITE_NAME — название сайта

%SITE_URL — адрес сайта

%USER_LOGIN — логин пользователя

%PASSWORD — пароль (только в подтверждении регистрации)

%CONFIRM_LINK — ссылка на подтверждение регистрации/восстановление пароля.

Классы и функции модуля

Класс nc_auth

Это основной класс для работы с модулем. Объект этого класса доступен в глобальной области видимости по имени

`$nc_auth`

так же объект этого класса можно получить так:

```
$nc_auth = nc_auth::get_object();
```

Методы класса:

auth_form (`$params = array()`, `$template = array()`)

Возвращает форму авторизации для неавторизованного пользователя и приветствие для авторизованного.

Если второй параметр метода с шаблонами отсутствует, то выводится форма авторизации по умолчанию на основе настроек, переданных в первом параметре.

Первый параметр представляет собой ассоциативный массив с ключами:

- `form_type` — тип формы:
 - `v` – вертикальная (по умолчанию)
 - `h` – горизонтальная
 - `t` – табличная
- `submit_name` — надпись на кнопке «Авторизоваться»
- `captcha_wrong` — надпись при неправильной введенной каптчи
- `login_wrong` — надпись при неправильно введенном логине
- `auth_text` — название блока «Авторизация»
- `login_text` — надпись «Логин»
- `reg_text` — надпись «Регистрация»
- `pass_text` — надпись «Пароль»
- `recovery_text` — надпись «Забыли пароль»
- `hide_register_link` — скрывать ссылку на регистрацию
- `login_save` — запомнить пароль:
 - `auto` — checkbox не показывается, но пароль запоминается
 - `checked` — показывается отмеченный checkbox
 - `none` – не показывается

Все параметры являются необязательными.

В зависимости от первого параметра, функция строит форму авторизации. Если ее внешний вид вас не устраивает, вы можете переопределить ее через шаблоны, передав их вторым параметром. В этом случае первый параметр будет проигнорирован.

Второй параметр также представляет с собой ассоциативный массив:

- `auth_form` — форма авторизации, в ней доступны следующие макропеременные:
 - `%form_id` — id формы
 - `%action` — action формы
 - `%register_link` — ссылка на регистрацию
 - `%recovery_link` — ссылка на восстановление пароля
 - `%login_save` — блок «запомнить меня»
 - `%token_form` — блок авторизации по токenu
 - `%vk_form` — блок авторизации через В Контакте
 - `%fb_form` — блок авторизации через Facebook
 - `%twitter_form` — блок авторизации через Twitter
 - `%openid_form` — блок авторизации через OpenID
- `login_save` — блок «запомнить меня». Если он не задан, используется код по умолчанию
- `vk_form` — форма авторизации через В Контакте, если не задан — код по умолчанию
- `fb_form` — форма авторизации через Facebook. если не задан — код по умолчанию
- `twitter_form` — форма авторизации через Twitter, если не задан — код по умолчанию
- `openid_form` — форма авторизации через OpenID, если не задан — код по умолчанию
- `token_form` — форма авторизации через токен, значения по умолчанию нет
- `authorized` — шаблон вывода приветствия, доступны следующие макропеременные:
 - `%login` — логин или имя пользователям
 - `%profile_link` — ссылка на профиль
 - `%exit_link` — ссылка на деавторизацию (выход)
 - `%messages` — информер о личных сообщениях
- `messages` — шаблон вывода сообщения о том, что нет новых сообщений (этот код как раз и заменяет макропеременную `%messages` в предыдущем шаблоне)

• `messages_new` — шаблон вывода сообщения о том, что новые сообщения есть (как и в предыдущем случае, этот код заменяет `%messages`), доступны следующие макропеременные:

- `%msg_url` — ссылка на личные сообщения
- `%msg_new` — количество новых сообщения

Все шаблоны, кроме `token_form`, имеют значения по умолчанию. Если вам надо изменить, допустим, только вывод информера о новых сообщениях, то в метод надо передавать только шаблон `messages_new`:

```
$nc_auth->auth_form( null, array ('messages_new' => 'Есть новые сообщения!'))
```

Примеры использования в макете дизайна:

```
".$nc_auth->auth_form() ."
```

```
".$nc_auth->auth_form( array('form_type' => 't') ) ."
```

```
".$nc_auth->auth_form( null, array('authorized' => "Вы авторизованы как  
%login") ) ."
```

`auth_links( $params = array(), $template = array() )`

Метод возвращает HTML-код со ссылками на регистрацию и авторизацию для неавторизованного пользователя и приветствие для авторизованного.

Параметры функции идентичны параметрам предыдущей функции, за исключением того, что можно переопределить шаблон вывода ссылок во втором параметра функции, в него можно добавить элемент с ключом `auth_link_form`:

```
$nc_auth->auth_links( null, array('auth_link_form' => 'Авторизация и  
Регистрация'))
```

В шаблоне по умолчанию ссылка на авторизацию показывает всплывающий слой, причем авторизация будет идти без перезагрузки страницы. Для показа всплывающего слоя используется плагин для `jquery simplemodal`, настроить его вид можно через CSS.

Управление рекламой на сайте

Модуль «Управление рекламой» позволяет организовывать рекламные кампании в определенных рекламных зонах на сайте, т.е. транслировать баннеры (GIF/JPEG, Flash, HTML) на страницах сайта с учетом статистики показов и кликов по баннерам, зонам, кампаниям и страницам показа. Статистические отчеты формируются в специальном интерфейсе управления модулем.

После успешной установки модуля вам будет предложено создать разделы «Баннеры», «Зоны», «Кампании», «Сценарии показов» и «Исключения» (ссылка «завершение установки»). К данным разделам будут автоматически подключены соответствующие компоненты. Если по каким-то причинам разделы не будут созданы автоматически, вы можете их создать самостоятельно, присвоив каждому соответствующий компонент.

Добавление баннера

В созданный системой раздел «Баннеры» с соответствующим компонентом вы можете добавлять ваши баннеры, предназначенные для трансляции. При добавлении графических баннеров необходимо указывать точный размер баннера. В случае если необходимый размер отсутствует в списке, добавьте его в разделе «Списки – Размеры баннеров». Также при добавлении баннера необходимо указать номер сценария показа. Вы можете поставить «1» (единицу), если не определили заранее собственный сценарий показа (сценарий с номером «1» добавлен при установке).

Сценарии показов

Сценарии показов определяют время и дни показа баннеров. Сразу после установки в компонент «Сценарии показов» добавляется сценарий «по умолчанию» с номером «1», определяющий показ баннеров круглосуточно, семь дней в неделю. Вы можете добавить любой другой сценарий показа и указывать его номер (ID) при добавлении баннера.

Исключения

Для определенного сценария показов можно указать запрет на показ баннеров в данном сайте, разделе, компоненте раздела или объекте. В

соответствующих полях при создании исключения необходимо указать ID соответствующих сущностей.

Добавление рекламной зоны

Рекламная зона – это место на сайте, где может транслироваться один или более баннеров определенного размера. При добавлении зоны вы можете определить место показа с точностью до конкретного объекта на сайте. Кол-во баннеров задается в функции `banner_rotate` вторым параметром. Если вам необходимо, чтобы баннеры в данной зоне ротировались случайным образом, укажите третий параметр `true` в этой функции.

Добавление рекламной кампании

Рекламная кампания связывает рекламные зоны и баннеры, определяя, в каких зонах какие баннеры транслировать, в какой период времени (дата начала и завершения кампании) а также количество показов, которые необходимо «открыть».

Рекламная кампания может связывать только баннеры и зоны определенного размера, поскольку очевидно, что невозможно показать, к примеру, в зоне с размером «100x100» баннер с размером «468x60».

В случае если вы не хотите, чтобы рекламная кампания пересекалась с кампанией аналогичного размера в аналогичных зонах (т.е. чтобы баннеры показывались по очереди), вы можете установить чекбокс «приостановить все пересекающиеся рекламные кампании».

В процессе рекламной кампании вы можете менять зоны показа и баннеры, транслируемые в них.

Настройка трансляции баннеров на странице

Для отображения баннеров на страницах сайта необходимо использовать функцию `banner_rotate($zone, $count, $rand)`, где `$zone` – номер зоны, `$count` – кол-во баннеров (можно не указывать, тогда будет отображаться один баннер), `$rand` – показывать выбранное кол-во баннеров случайным образом (по-умолчанию выключено, т.е. имеет значение `false`; укажите `true`, чтобы задействовать). При `$rand=true` баннеры выводятся случайным образом.

Разделитель между баннерами можно указать при создании баннерной зоны в поле «Разделитель между баннерами». Если в конкретном месте макета у вас имеются различные зоны определенного размера (в зависимости от раздела), то вы можете воспользоваться функцией определения конкретного размера зоны `banner_zone($size, $priority)`, где `$size` – номер элемента, определяющего размер зоны в списке «BannerSize», а `$priority` – приоритет зоны на странице (начиная с "0").

Пример использования этих функций в макете дизайна:

```
".banner_ rotate (banner_zone (2,0) ,3,true) ."
```

Данная функция будет возвращать HTML-код для 3-х баннеров 100x100, установленных для показа в рекламной зоне с соответствующим размером в текущем разделе, и ротирующихся случайным образом.

Статистические отчеты по баннерам, зонам, кампаниям и страницам показа можно просматривать в разделе «Модули – Управление рекламой».

Генерация статистики

Для создания статистики достаточно зайти на страницу модуля в разделе «Модули» системы администрирования. В системе предусмотрена автоматическая генерация статистики через скрипт, который помещен в «Управление задачами». Данный скрипт поможет Вам, если вы достаточно редко просматриваете статистику, и чтобы избежать ее переизбытка, после которого скрипт будет не в состоянии обработать такое кол-во записей.

Для настройки скрипта необходимо по FTP отредактировать файл `cron.php` в папке с модулем `/netcat/modules/banner`, в самом начале файла удалить строку с «`exit;`», после чего чуть ниже в строке «`$check="test";`» вместо `test` задать ключевое слово, лучше латинскими буквами. Слово нужно для предотвращения несанкционированного доступа к скрипту. Далее в разделе «Управление задачами» системы администрирования отредактируйте ссылку «`/netcat/modules/banner/cron.php?param=test`», указав вместо слова `test` выбранное вами ранее ключевое слово.

ZONE TABLE	Номер компонента "Banners: Зоны"	Число
SCRIPT TABLE	Номер компонента "Banners: Сценарии показов"	Число
BANNER TABLE	Номер компонента "Banners: Баннеры"	Число
CAMPAIGN TABLE	Номер компонента "Banners: Рекламные кампании"	Число
SCRIPT_EXCLUSION_TABLE	Номер компонента "Banners: Исключения в сценариях"	Число

Используемые функции:

- **banner_rotate (\$zone, \$count, \$rand)**

Данная функция возвращает HTML-код выбранного кол-ва баннеров, участвующих в ротации в указанной рекламной зоне (\$zone) в данной рекламной кампании. При \$rand=true баннеры выводятся случайным образом. Разделитель между баннерами можно указать при создании баннерной зоны в поле «Разделитель между баннерами».

- **banner_zone(\$size, \$priority)**

Функция определяет номер актуальной зоны с размером \$size для текущего сайта, раздела, компонента раздела или объекта. \$size – номер элемента в списке «BannerSize», определяющий размер баннера, а \$priority – приоритет зоны на странице (начиная с "0").

- **banner_url(\$id)**

Возвращает ссылку для баннера с номером \$id.

Интернет-магазин

Введение

Данный модуль позволяет организовать полнофункциональный интернет-магазин при помощи системы управления сайтами NetCat. Возможности модуля:

- интеграция с «1С:Бухгалтерия» (получение информации о номенклатуре, экспорт заказов в 1С в формате CommerceML);
- неограниченное количество типов товаров, продаваемых через Интернет-магазин (каждый тип товара может иметь свой особый набор свойств);
- возможность выбора различных цен на товары (розничная, мелкооптовая, оптовая и т.п.) для разных групп пользователей;
- ведение корзины товаров, оформление заказа с оповещением менеджеров организации о поступившем заказе по электронной почте или посредством SMS;
- поддержка различных валют, возможность указания цен на товары в разных валютах с их последующей автоматической конвертацией в базовую валюту магазина;
- автоматическое получение курсов валют Центрального Банка;
- гибкая система скидок и наценок на отдельные товары или на всю стоимость корзины;
- прием платежей через платежные системы Яндекс.Деньги, Webmoney в режиме «онлайн», а также по пластиковым картам Visa, Mastercard, JCB, DCL (при подключении к системе электронных платежей ASSIST);
- экспорт информации о товарах в формате YandexML для размещения на таких торговых площадках как <http://market.yandex.ru/>
- выбор варианта доставки товара покупателю с расчетом стоимости доставки;
- возможность указать особую ставку НДС для отдельных товаров;
- оформление документов для покупателя (счета, квитанции для оплаты через банк и т.п.);
- создание списка популярных товаров.

Установка модуля

Модуль может быть установлен на сайт с системой NetCat версии 3.5 и выше.

Если вы обладаете редакцией системы NetCat с уже установленным Интернет-магазином (E-Commerce, Extra), переходите к шагу 2.

1. **Если Вы приобрели модуль отдельно от системы**, загрузите имеющийся у Вас файл с дистрибутивом модуля через форму на странице «Инструменты» - «Установка модуля».

После первоначальной установки модуля в систему необходимо установить Интернет-магазин на ваш сайт. Для этого нажмите на ссылку «Завершить установку» в строке «Интернет-магазин» на странице «Модули». Укажите, на какой сайт вы хотите добавить Интернет-магазин (если вы хотите добавить магазин на несколько сайтов, повторите нижеописанную процедуру для каждого сайта).

2. Перейдите к редактированию настроек магазина (раздел «Каталог товаров»):

–Укажите реквизиты вашей компании. Эти данные будут использоваться для формирования счетов и квитанций на оплату, для подтверждения заказа по электронной почте и т.п.

–Укажите адрес, с которого будут отправляться сообщения, формируемые Интернет-магазином (например, письмо клиенту, подтверждающее оформление заказа).

–Укажите основную валюту магазина, в которой будут оформляться документы и рассчитываться окончательная стоимость товаров.

–Если вы хотите, чтобы при конвертации из одной валюты в другую добавлялся дополнительный процент, укажите его.

–На электронный адрес менеджера магазина будет высылаться оповещение о поступившем заказе. После окончания установки Интернет-магазина вы сможете указать менеджеров, ответственных за отдельные разделы каталога.

–Введите ставку НДС. Если товары, продаваемые через ваш магазин, не облагаются НДС, укажите «0»; если все или большинство продаваемых Вами товаров облагаются НДС 10%, укажите «10», иначе – «18». После установки при необходимости вы сможете указать ставку НДС для отдельных товаров.

После этого установите курсы валют (в разделе «Валюты»).

Если курсы валют не установлены, то при пересчете цен в данной валюте будет использован курс ЦБ РФ.

Для установки Интернет-магазина на другие сайты, имеющиеся в системе, откройте страницу «Модули - Интернет-магазин - Установка модуля на сайт». Для того, чтобы оформлять заказ могли только зарегистрированные на сайте пользователи, у компонента «Заказ» в разделе «Netshop» установите права на добавление для зарегистрированных пользователей.

Компоненты товаров

В Интернет-магазине, разработанном с использованием системы управления сайтами NetCat, может быть представлено неограниченное количество различных типов товаров. Каждый компонент типа товара может иметь свой набор свойств, однако *обязательно* должен содержать следующие поля:

- Name: название товара
- ItemID: артикул товара
- Price: основная цена товара
- Currency: валюта основной цены товара
- ImportSourceID: служебное поле, используется при импорте данных из внешних источников (например, 1С)

Все компонент товаров должны находиться в группе компонентов "Netshop". При создании нового компонента товаров укажите, что вы хотите создать его на основе существующего компонента «Товар».

Совет: не удаляйте поля из компонента «Товар», который поставляется с модулем «Интернет-магазин».

Цены для разных групп пользователей

Для разных групп пользователей вы можете назначать различные цены. По умолчанию (а также для незарегистрированных на сайте посетителей) используется колонка "Price". Если у вас нет необходимости иметь несколько различных типов цен, вы можете пропустить данный раздел Руководства. Для того чтобы добавить новый тип цены, во *всех* компонентах товаров необходимо ввести две новые колонки:

- PriceName, где Name – ключевое слово цены (например, PriceOpt или PriceDealer): цена единицы товара, поле типа «число с плавающей запятой»

-CurrencyName, где Name – ключевое слово, использованное для колонки «Цена»: валюта, в которой указана цена. Тип поля – «список», формат – «ShopCurrency».

Соответствия типов цен группам зарегистрированных пользователей задаются путем создания нового компонента «Цены для разных групп пользователей» в вашем Интернет-магазине. Например, чтобы для всех пользователей, являющихся членами группы с идентификатором «5» (идентификатор группы пользователей вы можете посмотреть в разделе «Пользователи» - «Группы пользователей» администраторской части NetCat), отображалась цена для дилеров, вам нужно создать новый компонент «Цены для разных групп пользователей» и указать ID группы пользователей (5) и название поля (PriceDealer).

Примечание: если вы хотите, чтобы группы пользователей (а также, соответственно, цены и доступные скидки) назначались автоматически при регистрации пользователей, обратитесь к разработчику вашего сайта. Как реализовать данную функциональность, указано в «Руководстве разработчика».

Минимальная цена

Особым типом цены является «Минимальная цена» - сумма, ниже которой цена на товар не может быть ни при каких условиях. Если при расчете скидок на данный товар была достигнута минимальная цена, то цена со скидкой будет равна минимальной, а дальнейшие скидки на товар применяться не будут. При импорте информации из 1С (см. ниже) вы можете указать соответствие поля «Минимальная цена» одному из типов цен (например, закупочной).

Скидки

В модуле «Интернет-магазин» системы NetCat имеются два уровня скидок: скидки на товар и скидки на всю сумму корзины. Первыми применяются скидки на товар, затем, соответственно, скидки на всю сумму заказа.

Чтобы создать скидку, создайте новый компонент «Скидка» в Интернет-магазине. Каждая скидка должна иметь понятное название с указанием размера скидки, например: «Скидка 5% при покупке на сумму более 1000 руб.».

Вы можете указать период действия скидки, типы цен, к которым данная скидка применяется, а также группы пользователей, которые могут ею воспользоваться. Если какое-либо из данных полей не заполнено, то скидка не имеет ограничения по этому параметру (т.е. ее срок действия не ограничен, она применяется ко всем типам цен и распространяется на все группы пользователей).

Одновременно несколько типов цен или групп пользователей могут быть выбраны мышью, при этом на клавиатуре необходимо удерживать клавишу «Ctrl».

Примечание: скидки применяются на всю сумму товара или покупки, включая НДС.

Скидки на отдельные товары

Для *скидок на отдельные товары* также существует возможность выбрать отделы магазина, в которых действует скидка, или отдельные товары. Несколько отделов или товаров могут быть выбраны мышью, при этом на клавиатуре необходимо удерживать клавишу «Ctrl».

Совет: если вы хотите установить скидку на отдельные товары, сначала выберите раздел (или несколько разделов), в котором находятся эти товары, а затем нажмите «Выбрать товары». В появившемся списке будут товары только из выбранного вами раздела.

Если группа товаров или раздел магазина не выбран, данная скидка применяется ко всем товарам.

Поле «Преобразование» описывает, какие изменения стоимости или количества товара в корзине должны быть произведены. Например, при описании скидки в 5% укажите преобразование «Стоимость умножить на 0.95». Размеры скидок и стоимость товара указываются в основной валюте Интернет-магазина.

Если вы хотите, чтобы скидка применялась при определенных условиях, например, только в случае если общая стоимость товара (т.е. произведение цены товара и его количества в корзине) больше какой-то суммы, необходимо заполнить поле «Условие».

Пример 1: Скидка 5% при покупке товара одного наименования на сумму более 1000.

Условие: $[TotalPrice] > 1000$

Преобразование: $[TotalPrice]$ (умножить) 0.95

Пример 2: Скидка 100 рублей при покупке товара одного наименования на сумму более 1000.

Условие: $[TotalPrice] > 1000$

Преобразование: $[TotalPrice]$ (вычесть) 100

Пример 3: «При покупке четырех единиц товара – пятая бесплатно»

Условие: $[Qty] = 4$

Преобразование: $[Qty]$ (равно) 5

Примечание: при добавлении подобной скидки Вы, возможно, захотите также добавить скидку «Пять единиц товара по цене четырех»:

Условие: $[Qty] = 5$

Преобразование: $[TotalPrice]$ (вычесть) $[ItemPrice]$

Если скидка изменяет количество товара, то, соответственно изменившемуся количеству, будет рассчитана новая стоимость единицы товара (цена = стоимость / количество), которая будет отображаться в корзине на сайте и в формируемых счетах на оплату. При этом цена не может опуститься ниже минимального уровня, если он задан для данного товара.

Если для товара действуют несколько скидок, они будут применяться одна за другой в порядке убывания их приоритета до тех пор, пока не будет достигнуто одно из условий:

- 1) у скидки установлен флаг «Дальнейшие скидки не применяются»;
- 2) достигнута минимальная стоимость товара;
- 3) применены все возможные для данного товара скидки.

Скидки на всю сумму заказа

Второй вариант скидок – скидки на всю сумму заказа. Для таких скидок «Стоимость $[TotalPrice]$ » означает стоимость всех товаров в корзине (с учетом примененных товарных скидок), а $[Qty]$ – количество различных наименований товаров в магазине.

Пример: скидка 10% при покупке товаров на общую сумму более 1000

Условие: [TotalPrice] > 1000

Преобразование: [TotalPrice] (умножить) 0.9

Если компоненты товаров в вашем магазине содержат дополнительные поля, то вы можете в квадратных скобках указать их название. Тогда в качестве условия будет выступать сумма по данному полю по всем товарам (пример см. в разделе «Наценки»).

При наличии нескольких удовлетворяющих условиям скидок на сумму заказа, они будут применяться одна за другой в порядке убывания их приоритета до тех пор, пока не будет достигнуто одно из условий:

- 1) у скидки установлен флаг «Дальнейшие скидки не применяются»;
- 2) сумма заказа меньше суммы минимальных цен товаров;
- 3) применены все возможные для данного товара скидки.

Скидки в зависимости от суммы предыдущих покупок пользователя

Вы можете использовать псевдо-переменную [PrevOrdersSum] в поле «Условие» для того, чтобы задавать скидку в зависимости от суммы всех завершенных заказов данного пользователя. Данное условие может быть применено как к отдельным товарам, так и к корзине в целом.

Пример: скидка 5% покупателям, купившим товаров на сумму более 100000

Условие: [PrevOrdersSum] > 100000

Преобразование: [TotalPrice] (умножить) 0.95

Для того, чтобы срабатывало условие [PrevOrdersSum] необходимо указать в параметре PREV_ORDERS_SUM_STATUS_ID в настройках модуля «Интернет-магазин» идентификаторы соответствующих статусов в списке Статусы заказов (ShopOrderStatus).

Наценки

При помощи механизма скидок вы также можете указывать *наценки* на товар или сумму заказа. Например, если в компонентах товаров у вас имеется поле «Вес в килограммах (Weight)», и Вы хотите сделать наценку 5% при покупке товаров общим весом более 10 кг, Вы можете создать «скидку для всей корзины», указав условие «[Weight] > 10» и преобразование «[TotalPrice] умножить на 1.05». Будьте внимательны при указании приоритета скидок и наценок, так как если больший приоритет будет иметь скидка с флагом «Дальнейшие скидки не применяются», данная наценка не будет применена.

Валюты

Каждый Интернет-магазин может иметь несколько валют, в которых указаны цены на товары. Одна из валют принимается в качестве *основной*. В ней ведутся расчеты стоимости покупки, указаны скидки, стоимость доставки и наценка за способы оплаты.

При оформлении заказа стоимость товаров будет пересчитана в основную валюту Интернет-магазина по курсу, указанному для данной валюты. В случае если курс не задан вручную, будет использован курс ЦБ. При каждом пересчете из одной валюты в другую к сумме будет добавлен процент, указанный в настройках Интернет-магазина.

Обратите внимание, что иногда возможны ситуации, когда преобразование валют происходит несколько раз, и при каждом пересчете будет добавлены дополнительные проценты. Например, при формировании квитанций для оплаты через Сбербанк, сумма в них всегда указывается в российских рублях. Если цена товара указана в евро, при этом основной валютой магазина (т.е. валютой, в которой рассчитывается сумма заказа) является доллар США, то при расчете цены на товар дополнительный процент будет добавлен два раза (при пересчете из евро в доллары и из долларов в рубли).

Предупреждение: будьте внимательны при смене основной валюты. Помните, что скидки, наценки, стоимость доставки указываются в основной валюте, поэтому при изменении этого пункта необходимо изменить их значения.

Добавление новой валюты

Для добавления новой валюты:

1. Убедитесь, что в списке валют («Разработка» - «Списки» - «Валюта») присутствует трехбуквенный код валюты (коды валют должны быть указаны в соответствии с международным стандартом ISO 4217-90 и Общероссийским классификатором валют ОК 014-94).
2. В Вашем Интернет-магазине добавьте компонент «Валюта».
3. Укажите курс валюты по отношению к рублю (независимо от того, какую валюту Вы используете в качестве основной). Если поле «Курс» не заполнено, данные будут получены автоматически с сервера ЦБ.
4. Введите сокращенное наименование валюты, например, «руб» (без точки).
5. В поле «Полное наименование валюты» через запятую укажите парадигму склонения названия валюты, например: «рубль, рубля, рублей» (один рубль, два рубля, пять рублей). Аналогично заполняется поле «Наименование дробной части валюты» («копейка, копейки, копеек»).
6. Вы можете настроить для данной валюты также правила форматирования чисел:
 - а. Порядок вывода суммы (%s) и сокращенного наименования валюты (#), например, сумма 10 рублей будет отформатирована следующим образом:
"%s #" – «10 руб»
"# %s" – «руб 10»

- б.Количество знаков после запятой, которые будут учитываться и выводиться при расчетах в данной валюте.
- с.Разделитель целой и дробной частей, разделитель групп разрядов.

Налоги

Все цены в Интернет-магазине должны быть указаны с учетом НДС. Обратите внимание, что скидки и стоимость оплаты рассчитываются исходя из полной суммы, включающей НДС.

Если некоторые товары, продаваемые через ваш Интернет-магазин, имеют особую ставку НДС (10%), Вы можете указать это в поле «НДС» в карточке данного товара.

Варианты доставки и оплаты

Чтобы добавить новый вариант доставки или оплаты, создайте соответствующий компонент в Вашем Интернет-магазине.

При описании вариантов доставки и оплаты возможно указать условие, по аналогии с условиями скидок для всей корзины. При этом используются следующие обозначения:

- [TotalPrice] – общая стоимость заказа.
- [Qty] – количество наименований в корзине.

Названия других полей, взятые в квадратные скобки: сумма значений по данному полю для всех товаров.

Пример: Доставка почтой возможна только в случае, если общий вес товаров меньше 1 кг. Условие: «[Weight] < 1»

Если условие не задано, данный способ доставки или оплаты возможен во всех случаях.

При необходимости сделать наценку при определенном способе доставки или оплаты, укажите множитель и/или наценку в абсолютных числах. Стоимость доставки или оплаты рассчитывается по формуле: стоимость покупки с учетом скидки * множитель + наценка.

Пример: Доставка товаров на сумму меньше 500 руб. стоит 150 руб.

Название: «Доставка по городу при покупке менее 500 руб.»

Условие: [TotalPrice] < 500

Множитель стоимости: (пусто)

Наценка: 150

Вы можете временно сделать недоступным определенный способ оплаты или доставки, нажав на кнопку «выключить» рядом с данным способом (или убрав флаг «включить» в карточке объекта).

Подключение платежных систем

Подключившись к системе электронных платежей ASSIST (<http://www.assist.ru>), вы сможете получать оплату по пластиковым картам Visa, EuroCard/MasterCard, DinersClub, JCB. Кроме того, модуль «Интернет-магазин» позволяет формировать счета на оплату в платежных системах «Webmoney» и «Яндекс-деньги» и получать оплату через данные системы.

ASSIST

Описание процедуры регистрации магазина в системе электронных платежей ASSIST и требования к Интернет-магазину приведены на сайте <http://www.assist.ru>.

После регистрации введите полученный вами идентификатор в поле «Идентификатор в ASSIST» в настройках Интернет-магазина. Необходимо создать две страницы на сайте (с уведомлениями об успешном и неуспешном платежах) и указать их адрес (вместе с названием протокола <http://> и именем сайта) в соответствующих полях в настройках Интернет-магазина.

Примечание: при оплате через ASSIST основной валютой магазина должна быть RUR, USD или EUR.

Результаты транзакций по пластиковым картам вы будете получать по электронной почте на адрес, указанный при регистрации в Assist.

Webmoney

Для получения оплаты через систему WebMoney вам необходимо получить персональный аттестат Webmoney. Процедура получения персонального аттестата описана на странице <https://passport.webmoney.ru/asp/AtstInfo.asp>. На странице WebMoney «Настройка» (<https://merchant.webmoney.ru/>) необходимо указать следующие настройки:

- Result URL:
http://www.ИМЯ_САЙТА.ru/netcat/modules/netshop/post.php?system=webmoney
- Success URL: страница на вашем сайте, куда будет переведен браузер покупателя в случае успешного выполнения платежа (например, с текстом, выражающим благодарность за покупку)
- Fail URL: страница на вашем сайте, куда будет переведен браузер покупателя в случае если платеж не выполнен
- Метод вызова Success URL, метод вызова Fail URL: POST
- Метод формирования контрольной подписи оповещения о платеже: MD5
- Тестовый режим: установите тестовый режим, пока не убедитесь, что оплата через платежную систему проходит правильно.
- Активность: да
- Secret Key: введите секретный ключ. Ключ должен быть известен только Вам.
- Высылать Secret Key...: нет
- Позволять использовать URL, передаваемые в форме: нет
- Передавать параметры в предварительном запросе: нет
 - В компоненте раздела «Настройки Интернет-магазина» необходимо указать следующие настройки:
- «Webmoney: кошелек продавца» - кошелек, на который будет приниматься оплата. **Кошелек должен быть в основной валюте Вашего магазина.**
- «Webmoney: secret key» - секретный ключ в системе WebMoney.

Пока установлен тестовый режим в настройках системы WebMoney, перевод денег на ваш счет не производится.

В случае успешной оплаты, в поле «Оплата» у заказа записываются дата и реквизиты платежа в системе WebMoney.

Яндекс-деньги

Для получения оплаты через систему «Яндекс-деньги» требуется подать заявку на подключение к системе (<http://money.yandex.ru/shop.xml?scid=936>). В заявке укажите желаемую схему подключения – «ЦПП e-mail уведомления».

После заключения договора вам вышлют форму для настройки Интернет-магазина. Скопируйте часть формы от первого тэга `input` включительно до начала текста «Номер заказа» в поле «Настройки "Яндекс-Деньги"»:

```
<input type="hidden" name="TargetCurrency" value="643">
<input type="hidden" name="currency" value="643">
<input type="hidden" name="wbp_InactivityPeriod" value="2">
<input type="hidden" name="wbp_ShopAddress" value="wn1.paycash.ru:8828">
<input type="hidden" name="wbp_ShopEncryptionKey" value="(ключ) ">
<input type="hidden" name="wbp_ShopKeyID" value="0123456789">
<input type="hidden" name="wbp_Version" value="1.0">
<input type="hidden" name="wbp_CorrespondentID" value="(идентификатор) ">
<input type="hidden" name="BankID" value="1001">
<input type="hidden" name="TargetBankID" value="1001">
<input type="hidden" name="PaymentTypeCD" value="PC">
<input type="hidden" name="ShopID" value="идентификатор проекта">
```

Результаты оплаты вы будете получать по электронной почте на адрес, указанный при регистрации в системе «Яндекс-деньги».

Редактирование заказов

После оформления заказа по электронной почте высылается заявка менеджеру Интернет-магазина, либо менеджерам отделов, к которым относятся заказанные покупателем товары (если они указаны). В случае если менеджеры обнаружили неточность в заказе или хотят скорректировать стоимость товаров или доставки, а также скидки, то это можно сделать, отредактировав компонент заказа в вашем Интернет-магазине.

Если к товарам в заказе были применены какие-либо скидки, подробности вы можете узнать, наведя курсор мыши на знак «?» около суммы скидки. Обратите внимание, что при редактировании заказа дробная часть чисел должна отделяться от целой точкой (а не запятой).

Для удобства вы можете указать размер скидки, стоимость доставки и наценку за способ оплаты в процентах (например, «10%»). При сохранении изменений, внесенных в заказ, относительные значения будут пересчитаны в абсолютные.

Добавление полей в компонент «Заказ»

При необходимости (например, если вы хотите узнать реквизиты покупателя) Вы можете добавить любые поля в компонент «Заказ». Эти поля будут отображаться при оформлении и редактировании заказа, а также в уведомлении менеджерам о поступившем заказе. Если существуют поля с тем же именем в системной таблице «Пользователи», то каждый раз при оформлении заказа зарегистрированными пользователями значения полей из таблицы «Пользователи» будут автоматически перенесены в диалог «Оформление заказа».

Настройка шаблонов писем

Существует возможность настроить вид письма, высылаемого покупателю после оформления заказа. Для этого необходимо отредактировать шаблон писем «OrderConfirmation». В шаблоне вы можете указать следующие поля:

%CUSTOMER_...%	Поля с префиксом «CUSTOMER_» соответствуют полям компонента заказа, например: %CUSTOMER_CONTACTNAME% вставляет в текст письма значение поля ContactName данного заказа
%SHOP_...%	Поля с префиксом «SHOP_» содержат соответствующие настройки Интернет-магазина, например, %SHOP_PHONE% будет заменен в письме на значение поля Phone, указанного в настройках магазина
%CART_CONTENTS%	Содержимое корзины (наименование и артикул товара, цена, количество, стоимость с учетом товарной скидки)
%CART_DISCOUNTS%	Скидки на всю сумму покупки, примененные при оформлении заказа
%CART_DELIVERY%	Условия и стоимость доставки
%CART_PAYMENT%	Выбранный покупателем способ оплаты заказа
%CART_COUNT%	Количество наименований (например, «5 товаров»)
%CART_SUM%	Сумма заказа
%ORDER_ID%	Номер заказа

Популярные товары

Вы можете разместить на страницах Интернет-магазина список наиболее популярных товаров. Как это сделать, описано в руководстве разработчика, прилагаемом к данному модулю.

При помощи полей «Множитель для списка «Лидеры продаж» и «Надбавка в списке «Лидеры продаж» можно регулировать позицию отдельных товаров в данном списке. Популярность товара рассчитывается по формуле: «количество единиц проданного товара * множитель + надбавка».

Оповещение о заказе разным менеджерам

Каждый раздел Интернет-магазина может иметь своего менеджера, которому по электронной почте будет высылаться оповещение об оформлении заказа. Если заказ содержит товары из разделов, за которые отвечают разные менеджеры, каждый из них получит оповещение со списком товаров из его раздела.

Для того чтобы указать адрес электронной почты менеджера раздела и его подразделов, добавьте в этот раздел компонент «Раздел магазина», затем добавьте объект, в котором укажите e-mail для оповещения о заказе.

Если у раздела, к которому относятся товары в заказе, и у его родительских разделов не указан адрес электронной почты, оповещение высылается на адрес, указанный в настройках Интернет-магазина.

Импорт данных из 1С

Импорт данных из 1С позволяет с минимальными затратами времени создать структуру Интернет-магазина, а в дальнейшем – обеспечивать актуальность цен на товары в магазине.

Перед выгрузкой информации из 1С, пожалуйста, убедитесь, что на сайте созданы все необходимые компоненты товаров (как добавить новый компонент и требования к нему – см. выше) и *каждый* из них имеет одинаковый набор колонок цен (см. «Цены для разных групп пользователей»).

Первая обработка номенклатуры 1С

1. В меню 1С выберите пункт «Сервис» - «Обмен данными в формате CommerceML» - «Выгрузка пакета коммерческих предложений».
2. Выберите цены и разделы каталога, которые вы хотите выгрузить на сервер.
3. Отметьте пункт «Выгружать каталог товаров».

4.Нажмите кнопку «Подбор номенклатуры». На вкладке «Выбор элементов» нажмите кнопку «Выбрать показываемые реквизиты», отметьте поля «Основное представление», «Группа (родитель)», «Артикул», «Базовая единица измерения». Нажмите «ОК».

5.Создайте файл выгрузки на вашем (локальном) компьютере.

6.Если вы еще не импортировали данные из 1С на сайт, зайдите в раздел «Настройки» – «Настройки модулей» – «Интернет-магазин» – «Импорт данных в формате CommerceML».

Внимание! Для корректной работы в вашем браузере должна быть включена поддержка JavaScript.

7.Если несколько сайтов под управлением данной копии NetCat имеют Интернет-магазин, укажите тот из них, куда следует добавить данные из 1С. **Будьте внимательны!** Если вы неправильно укажете магазин, туда будут добавлена номенклатура из 1С.

8.Введите название нового источника, например, «Номенклатура 1С». Нажмите кнопку «Далее».

9.Укажите файл, сформированный 1С в пункте 1.

10.Если вы хотите, чтобы в дальнейшем те позиции, которые будут удалены из номенклатуры 1С, продолжали отображаться на сайте, в пункте «Что делать с позициями, которых нет в источнике», выберите «оставить как есть».

11.Укажите соответствие групп цен в 1С и в Интернет-магазине. Если на сайте имеется несколько компонентов товаров, система попросит также указать тип компонента для каждого несуществующего на сайте раздела.

12.После первой обработки номенклатуры 1С система выдаст вам ссылку, по которой в дальнейшем вы сможете загружать информацию на сайт непосредственно из 1С.

Импорт из 1С8

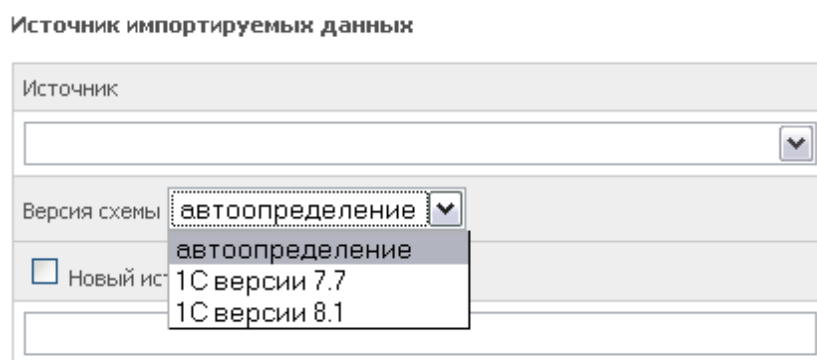


Рис.1

Импорт из 1С8 (CommerceML 2.03 и выше), происходит аналогично импорту из предыдущей версии. Инструментарий находится в административной части системы, в панели управления модуля ("Настройки" — "Список модулей" — "Интернет магазин" — "Импорт данных в формате CommerceML").

Единственное требование к импорту - файл должен быть выгружен из 1С в кодировке UTF-8. В меню импорта добавлено поле "Версия схемы". В обычных ситуациях следует оставить значение этого поля "автоопределение". Если же система по каким либо причинам не может распознать файл импорта, следует явно указать версию 1С.

В версии импорта для 1С8 предусмотрена загрузка изображений. Соответствие полей указывается так же, как и в предыдущей версии. Файлы с изначальной структурой директорий должны находиться в папке /netcat/tmp/.

Выгрузка информации из 1С

После того как вы выполните все пункты из раздела «Первая обработка номенклатуры 1С» и получите ссылку для прямой загрузки информации на сайт, вы можете выгружать информацию на сайт непосредственно из 1С. Для этого откройте 1С, затем:

1. В меню 1С выберите пункт «Сервис» - «Обмен данными в формате CommerceML» - «Выгрузка пакета коммерческих предложений».
2. Выберите цены и разделы каталога, которые Вы хотите выгрузить на сервер.
3. Отметьте пункт «Выгружать каталог товаров».
4. Нажмите кнопку «Подбор номенклатуры». На вкладке «Выбор элементов» нажмите кнопку «Выбрать показываемые реквизиты», отметьте поля «Основное представление», «Группа (родитель)», «Артикул», «Базовая единица измерения».
5. Отметьте пункт «Отправить на сайт» и нажмите на многоточие (...)
6. В диалоговом окне нажмите «Новая строка», введите наименование сайта. В поле «Адрес» укажите: ссылку, полученную при ручной загрузке файла через сайт. Поля «Имя пользователя» и «Пароль доступа» оставьте пустыми.

Внимание! Если вы создаете новые разделы в 1С, они не будут отображены на сайте до тех пор, пока вы не загрузите данные через сайт, как указано в разделе «Первая обработка номенклатуры 1С».

Выгрузка заказов в 1С

Для выгрузки информации о заказе нажмите в соответствующем разделе на ссылку «Выгрузить в CommerceML (1С)» рядом с данным заказом. В 1С выберите меню «Сервис» - «Обмен данными в формате CommerceML» - «Просмотр и загрузка документа» и укажите полученный вами файл, а затем нажмите кнопку «Показать». Если напротив названий товаров или услуг имеется надпись «не идентифицирован», нажмите кнопку «Просмотр каталога товаров» и двойным нажатием на строки с неидентифицированными позициями добавьте их в номенклатуру 1С.

После нажатия на кнопку «Загрузить» в 1С будет создана заявка, содержащая соответствующие товарные позиции. Если на сайте не требуется регистрация пользователей для оформления заказа, заявка может содержать неверную ссылку на покупателя; в этом случае необходимо в 1С указать контрагента вручную.

Если вы хотите, чтобы в заявке проставлялось название организации (а не контактное лицо) в качестве названия контрагента, создайте в компоненте «Заказ» поле с названием «CompanyName».

Выгрузка предложений в Яндекс.Маркет

Чтобы выгружать каталог товаров на Яндекс.Маркет, ваш магазин должен быть зарегистрирован в данной системе. Условия участия описаны на странице <http://market.yandex.ru/help.xml?id=86972>.

При экспорте информации в Yandex.Market укажите адрес <http://www.yourshop.ru/netcat/modules/netshop/export/yandexml.php>

Выгрузка предложений в TorgMail

Выгрузка товаров доступна по адресу <http://www.yourshop.ru/netcat/modules/netshop/export/torgmail.php>

Настройка компонентов: вывод информации о товарах

В шаблоне вывода информации о товарах (список или подробности) в *Системных настройках* должна присутствовать запись:

```
if (!$curPos)
{
    $shop = new NetShop($_POST['cart']);
    extract((array)$GLOBALS["NETSHOP"]);

    $query_select = "a.$shop->PriceColumn, a.$shop->CurrencyColumn";
    $result_vars = "\$Price, \$Currency";
}
```

Для вывода информации о товарах в Интернет-магазине используются стандартные переменные. Цена, действующая для группы текущего пользователя, находится в переменной `$Price`, а валюта этой цены — в переменной `$Currency`.

Для форматирования цен согласно настройкам валют используйте метод `FormatCurrency()` объекта `$shop`:

```
Цена " . $shop->FormatCurrency($Price, $Currency) . "
```

Метод `ConvertCurrency()` позволяет конвертировать сумму из одной валюты в другую по действующему курсу с добавлением дополнительного процента (если он указан в настройках магазина):

```
Цена в основнойв алютемагазина:" . $shop->ConvertCurrency($Price,
$Currency) . "
Цена в долларах " . $shop->ConvertCurrency($Price, $Currency, 'USD') . "
```

Для того чтобы можно было добавить товар в корзину, создайте в шаблоне следующую форму:

```
<form method='post' action='/netcat/modules/netshop/post.php'>
<input type='hidden' name='redirect_url' value='$GLOBALS[uri_path]' />
<input type='input' size='3' name='cart[$classID][$f_RowID]' value='1' />
<input type='submit' value=' в корзину'>
</form>
```

Настройка компонентов: целое и дробное количество товаров

Предусмотрена возможность использования как целого, так и дробного количества товаров. В зависимости от выбора типа поля `StockUnit` в компоненте "Магазин: Товар" целого или с плавающей точкой у вас можно будет указывать при покупке целое или дробное число.

В списке ("Разработка" - "Списки") "Единицы измерения" добавьте метры или другие единицы измерения, в которых вы будете измерять ваши товары. При добавлении товара новая единица измерения стане доступной.

Важно: Для использования обоих вариантов, когда есть одни товары, для которых необходимо указать целое, а для других дробное количество, - необходимо оставить стандартный компонент "Магазин: Товар" без изменений (для целых чисел) и создать на его основе другой для дробных.

Предустановленные переменные

При инициализации модуля определяется ряд значений в массиве `$GLOBALS["NETSHOP"]`.

Для удобства стандартные компоненты непосредственно после инициализации модуля содержат команду `extract((array) $GLOBALS["NETSHOP"])`, что позволяет использовать эти параметры «напрямую», в виде переменных:

<code>\$Netshop_TotalPrice</code>	Сумма товаров в корзине (отформатированная строка с указанием валюты). Сумму в виде числа можно получить через метод <code>\$shop->CartSum()</code>
<code>\$Netshop_ItemCount</code>	Количество различных наименований товаров в корзине (число)
<code>\$Netshop_CartContents</code>	Содержимое корзины (массив). Каждый элемент массива, помимо полей компонента товара, содержит следующие записи: • <code>Class_ID</code>: идентификатор компонента товара • <code>RowID</code>: строка «<code>[id_компонента][id_товара]</code>» • <code>Qty</code>: количество товара в корзине • <code>ItemPriceF</code>: отформатированная цена единицы товара с указанием валюты • <code>TotalPriceF</code>: отформатированная стоимость товара (цена * количество) с указанием валюты • <code>Discounts</code>: массив с информацией о примененных к товару

	скидках: 1.Sum: сумма скидки 2.SumF: сумма скидки (отформатированная строка) 3.Name: название скидки •Description: описание скидки PriceMinimum: логическая переменная (истинна, если при расчете скидки достигнута минимальная цена товара)
\$Netshop_CartDiscountSum	Сумма скидки на корзину (отформатированная строка с указанием валюты). Сумма скидки в виде числа: \$shop->CartDiscountSum()
\$Netshop_CartURL	Путь к «Корзине»
\$Netshop_OrderURL	Путь к компоненту раздела «Заказ»
\$Netshop_OrderLink	Ссылка на компонент раздела «Заказ»
\$Netshop_CartSum	Строка: - если корзина пуста: «Ваша корзина пуста» - если в корзине есть товары: «В корзине ... товаров на сумму ...»

Настройка вывода корзины товаров

Пример настройки внешнего вида корзины при помощи CSS можно посмотреть в предустановленном компоненте «Корзина».

Добавление списка популярных товаров

Чтобы добавить список популярных товаров, воспользуйтесь методом `GetBestsellers()` объекта `$shop`.

`$shop->GetBestsellers($type_ids, $number, $section=false)`

i.`$type_ids`: типы товаров через запятую, которые будут включены в список. Если ничего не указано – все типы товаров

ii.`$number`: количество товаров в списке

iii.`$section`: `Subdivision_ID` раздела, для которого будет построен список

Метод возвращает массив, каждый элемент которого содержит поля компонента товара.

```
";  
  
$list = $shop->GetBestsellers(); // все типы товаров, 5  
позиций, все разделы  
  
foreach ($list as $row)  
{  
    $result .= "<a href='$row[URL] '$row[Name]</a><br />";  
}  
  
$result .= "
```

Настройка вывода корзины товаров

Содержимого корзины доступно через массив `$Netshop_CartContents`. Что находится в этом массиве, а также другие полезные переменные см. выше, в разделе «Предустановленные переменные».

Пример альтернативного вывода корзины (упрощенный вариант для иллюстрации алгоритма):

```
";
```

```

extract ($GLOBALS['NETSHOP']);

foreach ($Netshop_CartContents as $item) {
    $result.= "<li><a href='$item[URL]'" . $item['Name'] . "</a>:"
        ". $item['Qty'] ." * " . $item['ItemPriceF'] ." = " .
$item['TotalPriceF'] . "</li>";
}

$result .= "</ol>
<p><b>ИТОГО: $Netshop_TotalPrice</b><br>
<a href='$Netshop_OrderURL'>Перейти к оформлению заказа</a></p>

```

Настройки модуля

В настройках модуля («Настройки» - «Настройки модулей» - «Интернет-магазин» - «Настройки») Вы можете изменить следующие параметры:

- **SECRET_KEY**: набор символов, который используется для формирования уникальных ссылок, например, для получения квитанций на оплату и т.п. Секретный ключ генерируется при установке модуля. Не меняйте секретный ключ магазина без явной необходимости.
- **RATES_DAYS_TO_KEEP**: количество дней, в течение которых будут храниться курсы валют ЦБ (0 – не удалять курсы валют).
- **PREV_ORDERS_SUM_STATUS_ID** — идентификаторы статусов заказов, перечисленные через запятую, для подсчета суммы предыдущих покупок пользователя. Рекомендуется 3,4.

Управление ссылками

Модуль автоматизирует управление ссылками на сайте – обмен ссылками, продажа и покупка ссылок. Основные функции модуля:

1. Обмен ссылками

- a. Возможность вносить ссылки как для администратора сайта, так и посетителей (администраторов других сайтов)
- b. Два режима работы: режим обязательного наличия ссылки (при отсутствии обратной ссылки на сайте партнера его ссылка может быть выключена или удалена) и режим редиректа (при отсутствии обратной ссылки ссылка на сайт партнера не прямая)
- c. Периодическая автоматическая проверка наличия обратных ссылок, автоматическое включение, выключение, удаление, перевод в режим редиректа и обратно
- d. Возможность организовать асинхронный обмен для двух сайтов (партнер размещает ссылку на сайт А, а ссылка партнера размещается на сайте Б)
- e. Автоматические регулярные уведомления администратора и партнеров о добавлении ссылки, изменении статуса ссылки
- f. Информация о сайтах-партнерах – ТИЦ Яндекса, Google PageRank; проиндексированные страницы, количество ссылок на сайт, наличие в каталогах и пр.
- g. Поиск по каталогу ссылок
- h. Фиксированные адреса для страницы со ссылкой
- i. Автоматический листинг списка ссылок по страницам
- j. Возможность организации стоп-листа (запрет на добавление определенных сайтов)
- k. Несколько вариантов кода собственной ссылки
- l. Неограниченное количество разделов, многоуровневый рубрикатор разделов
- m. Гибкая система настроек условий обмена

2. Продажа ссылок

- a. Возможность размещать ссылки на другие сайты на внутренних страницах и/или на титульной странице
- b. Автоматическое выключение ссылки по истечении срока продажи
- c. Автоматические уведомления партнеров о выключении ссылки

3. Покупка ссылок

- a. Автоматическое отслеживание наличия купленной ссылки, уведомление при ее отсутствии

Описание функциональности

В составе модуля поставляются 5 компонентов (см. ниже). Для каждого компонента в структуре первого сайта автоматически создаются разделы. Если на одной копии системы работают несколько сайтов, для них тоже можно организовать каталоги ссылок, вручную создав разделы и добавив соответствующие компоненты в созданные разделы.

Изначально раздел «Ссылки» выключен, в нем содержатся несколько подразделов. Они делятся на две группы: подразделы каталога (включены) и служебные разделы (выключены). Тематическими подразделами можно управлять при помощи стандартных инструментов NetCat – добавлять, удалять, изменять название, создавать подразделы 3, 4 уровня и т.д. Если вы создаете новый подраздел, добавьте в него компонент «Ссылки: каталог ссылок».

В служебных (выключенных) подразделах содержатся коды ссылок вашего сайта (для установки на сайтах-партнерах), коды проданных и купленных ссылок, а также список доменов стоп-листа (сайтов, запрещенных к обмену).

При добавлении ссылки модуль проверяет ее на соответствие условиям обмена (см. ниже – «Настройки модуля»). Если ссылка не удовлетворяет условиям, пользователю выводится предупреждение, если удовлетворяет – ссылка добавляется. Одновременно с добавлением посетителю и администратору отправляется письмо с информацией о добавленной ссылке (эту возможность можно отключить – см. «Настройки модуля»).

Проверка всех ссылок, расположенных в базе, может производиться как вручную, так и автоматически. По умолчанию проверка производится раз в сутки (периодичность можно изменить в разделе «Инструменты» - «Управление задачами» системы управления сайтами NetCat). В ходе выполнения проверки модуль «обходит» все страницы, на которых должны быть расположены ссылки, и в случае отсутствия «нашей» ссылки включает/удаляет ссылку партнера или переводит ее в режим редиректа (см. ниже – «Настройки модуля»). Если партнер не выполнял условия обмена (ссылка находилась в режиме редиректа или была выключена), но в ходе очередной проверки обнаружилось, что он стал их выполнять, ссылка переводится в обычный режим. При каждом изменении статуса ссылки партнеру высылается письмо (эта возможность может быть отключена).

Также в ходе проверки модуль отслеживает наличие купленных ссылок и выключает проданные ссылки (при окончании срока продажи).

На странице каталога ссылок доступны следующие возможности:

1. Просмотр списка ссылок как по категориям, так и в полном объеме
2. Поиск ссылок
3. Переход по подразделам каталога
4. Переход на страницы с условиями обмена и с кодами «наших» ссылок
5. Добавление ссылки

Начало работы

Для начала работы со ссылками вам необходимо внести информацию в служебные разделы. Также вы можете добавить несколько ссылок в каталог, заполнить список купленных и проданных ссылок (при наличии таковых). Ниже приведены описания всех компонентов.

Каталог ссылок

Компонент используется для хранения и отображения ссылок на сайты партнеров. В режиме просмотра для каждой ссылки отображается ее номер, название сайта, описание, URL и ссылка на ее статическую страницу. В режиме редактирования администратор может также просмотреть внешние характеристики этого сайта: тИЦ Яндекса, Google PageRank, количество проиндексированных страниц в разных поисковых системах, количество ссылок на сайт, установленные счетчики и пр.

Поля компонента:

- Название сайта
- URL сайта
- HTML-код ссылки на сайт
- URL страницы с «нашей» ссылкой
- E-mail администратора сайта
- Не обрабатывать ссылку роботом (доступно только администратору)

Последнее поле используется для особых случаев, например, в случае договоренности о взаимной ссылке. Также в компоненте присутствует несколько служебных полей, необходимых для работы модуля.

Коды ссылок

Компонент предназначен для ввода HTML-кодов «наших» ссылок. Если объектов в соответствующем разделе несколько, партнер может установить любой из них на свой выбор, например, кнопку или текстовую ссылку. На странице с кодами «наших» ссылок отображается внешний вид ссылки и поле `<textarea>`, для того чтобы партнер мог скопировать код ссылки прямо со страницы.

Поля компонента:

- Название сайта
- URL сайта
- HTML-текст ссылки

Обратите внимание, что содержимое первых двух полей не влияет на результаты проверки роботом и предназначены только для информативных целей.

Продажа ссылок

Компонент предназначен для хранения HTML-кодов проданных ссылок. Обратите внимание, что изначально раздел выключен. Его содержимое (сами ссылки) обычно выводят на страницах с высоким PageRank (например, на титульной странице). Выводить содержимое раздела на нужной странице можно при помощи стандартной функции `NetCat s_list_class()` (см. Руководство разработчика, Приложение №2).

Поля компонента:

- Название сайта
- E-mail партнера
- HTML-текст ссылки
- Дата отключения

Покупка ссылок

Данный компонент носит исключительно технический характер и не выводится на сайте. В ходе проверки робот модуля «обходит» указанные сайты и ищет на них указанные HTML-коды. В случае если коды отсутствуют, и время выкупа ссылки еще не наступило, администратору сайта отправляется письмо.

Поля компонента:

- Название сайта
- HTML-текст «нашей» ссылки
- E-mail партнера
- URL страницы с «нашей» ссылкой
- Дата окончания срока выкупа

Стоп-лист

Если вы не хотите меняться ссылками с какими-либо сайтами (например, это ваши конкуренты или они плохо зарекомендовали себя), вы можете добавить их в стоп-лист. Данный компонент также является техническим и обычно не отображается на сайте.

Поля компонента:

- Домен сайта (без префикса http://)
- Комментарий

Обратите внимание, что термин «обмен», который использован выше, не всегда обозначает обмен обратными ссылками между двумя сайтами. Модуль также поддерживает возможность асинхронного обмена двумя способами:

1. Ссылки партнеров располагаются на сайте А, а партнер должен ставить ссылку на сайт Б. В этом случае необходимо внести HTML-код ссылки на сайт Б в раздел «Коды ссылок».
2. Ссылки на «наш» сайт должны располагаться на сайте, отличном от того, на который ссылается наш сайт. Эта опция устанавливается в настройках модуля (см. ниже).

После внесения всех необходимых данных следует настроить модуль (см. ниже), произвести изменения в тематическом рубрикаторе ссылок

(стандартными средствами системы NetCat) и расположить ссылки на соответствующие страницы в макетах дизайна сайта.

Если раздел «Ссылки» должен располагаться в меню, его можно просто включить, проставив соответствующую галочку в настройках раздела. Если ссылка на страницу ссылок может располагаться в произвольном месте страницы, необходимо в соответствующем макете дизайна вручную ввести HTML-код примерно следующего содержания:

```
<a href='/links/'>Полезные ссылки</a>
```

Вывод содержимого компонента «Ссылки: продажа ссылок» на страницы/страницу сайта осуществляется при помощи функции `s_list_class()`.

Обратите внимание, что у всех сайтов, для которых производится обмен ссылками, должен быть указан домен (в настройках сайта).

Совет: без консультаций разработчика системы не изменяйте состав полей компонентов, которые поставляются с модулем.

Настройка модуля

Для настройки модуля управления ссылками зайдите в подраздел «Модули» раздела «Настройки» интерфейса администрирования NetCat и нажмите на название модуля. Первая страница содержит справочную информацию о ссылках и кнопку «Запустить проверку». Для произведения настройки модуля нажмите на соответствующую ссылку.

Экран «Настройки модуля»

Обмен ссылками может происходить в одном из двух режимов – обязательное наличие обратной ссылки и редирект при отсутствии обратной ссылки. Режим изменяется при помощи соответствующего переключателя (тип `radiobutton`).

В первом случае, если обратная ссылка не найдена, робот либо включает ее (если эта опция установлена), либо удаляет ее из базы. Если ссылка выключена, она удаляется через указанное количество дней.

Во втором случае ссылка на сайт (если обратной ссылки нет) может производиться через т.н. редирект, то есть имеет вид не <http://www.partnersite.ru>, а <http://www.oursite.ru/netcat/modules/linkmanager/redirect.php>? <http://www.partnersite.ru>. Поисковые машины в таком случае не считают эту ссылку внешней, и ее наличие не влияет на тИЦ сайта партнера. Также из HTML-кода ссылки партнера могут быть удалены все HTML-теги. В режиме редиректа модуль регулярно отправляет партнеру письмо с предложением разместить обратную ссылку (если в соответствующем поле настроек модуля не указан ноль).

Общие настройки для обоих режимов работы:

- Проверять наличие на сайте партнера полного текста ссылки (который введен в разделе «Коды ссылок») или только ссылки на текущий сайт (формат ссылки и ее описание партнер определяет сам)
- Где должна быть расположена обратная ссылка: на том же сайте, куда ссылается наш сайт, на любом другом сайте, или же это не принципиально.
- Разрешать ли добавление, если обратная ссылка с указанного сайта уже присутствует в базе
- Разрешать ли добавление, если в коде ссылки партнер указал адрес не только своего сайта, но и других сайтов
- Разрешать ли добавление, если «наша» ссылка расположена не на домене второго уровня (например, домен www.site.ru или site.ru является доменом второго уровня, а something.site.ru – третьего уровня)

Также вы должны указать адреса электронной почты для отправления автоматически генерируемых писем (адрес может быть ненастоящим, однако в этом случае Вы не сможете отслеживать неправильные адреса, которые вводят партнеры; также правила хорошего тона требуют, чтобы в шаблонах автоматически генерируемых писем было указание на то, что адрес не является настоящим и отвечать на автоматически генерируемые письма не нужно) и свой адрес, на который будут приходить автоматически генерируемые отчеты и уведомления (см. ниже).

Экран «Шаблоны писем»

При каждой проверке модуль генерирует и рассылает письма администратору сайта и партнерам. События, при которых происходит рассылка:

- 1.Добавление ссылки (администратору и партнеру)
- 2.Автоматическое включение/выключение ссылки (партнеру)
- 3.Автоматическое удаление ссылки (партнеру)
- 4.Автоматическое включение/выключение режима редиректа (партнеру)
- 5.Выключение проданной ссылки (партнеру)
- 6.Исчезновение купленной ссылки (администратору)

Также администратору отправляется отчет об итогах проверки.

Каждую рассылку можно отменить, выключив галочку в соответствующем блоке. Для каждого типа события шаблон письма может быть настроен – тема письма и его содержимое. Т.к. в теле письма должны быть указаны конкретные адреса и названия, в шаблонах используются макропеременные, которые при отправке автоматически заменяются на значения для каждой ссылки. Ниже приведены все макропеременные и их значения:

- %OurSite – домен (адрес) «нашего» сайта
- %LinkName – название сайта партнера
- %LinkURL – адрес сайта партнера
- %BackLink – URL страницы с обратной («нашей») ссылкой
- %LinkEditMode – ссылка на полный вывод ссылки в режиме администрирования
- %StaticLink – статический (постоянный) адрес страницы на «нашем» сайте со ссылкой партнера
- %AdminEmail – e-mail администратора сайта (Ваш e-mail)
- %OurCatalog – начальная страница каталога ссылок (URL)
- %DeleteTime – время удаления ссылки (для режима обязательного наличия ссылки, если включена опция «выключать ссылки при отсутствии обратной ссылки»)
- %LinkEmail – e-mail партнера
- %LinkTurnoff – дата запланированного выключения купленной ссылки

Обратите внимание, что все макропеременные чувствительны к регистру, т.е. Вы не можете написать %oursite вместо %OurSite.

Облако тегов

Облако тегов – один из неотъемлемых атрибутов порталов в стиле web 2.0 – блогов, сообществ и больших социальных ресурсов. Помимо эстетических составляющих, облако тегов выполняет функцию структурированного каталога с закладками. Принцип формирования такого облака предельно прост: при добавлении новости, публикации или сообщения пользователь указывает так называемые метки (или теги), которые, по его мнению, должны обобщить написанную им информацию, являться ключевыми словами к ней. Далее, для удобства, эти ключевые слова будем называть тегами.

В зависимости от частоты появления того или иного тега в каком-либо сообщении, складывается вес, или попросту его рейтинг. Из таких тегов с зависимостью размера шрифта от частоты его упоминания (веса) и формируется облако тегов. Т.к. наполнение сайта постоянно меняется, то количество тегов, их вес и визуальное представление облака тегов также постоянно колеблется, что, несомненно, придаёт частичку самостоятельности сайтам с таким функционалом.

Требования к компонентам

Для подключения функционала добавления тегов к сообщениям, следует выполнить некоторые действия по модификации компонентов. Допустим, требуется добавить теги в компонент новостей. Первое, что следует выполнить, это добавить поле **Tags** (тип поля - строка) в список полей компонента. В этом поле пользователи смогут через запятую перечислять нужные теги для сообщений. Следующие шаги направлены на добавление функций управления тегами в «действие после добавления», «действие после изменения» и «действие после удаления» компонента.

- В действии после добавления следует прописать вызов функции **nc_tag_add(\$sub, \$cc, \$msgID, \$f_Tags)**, подробное описание ниже. Данная функция позволит добавить теги в сообщение.
- В действии после изменения следует прописать вызов двух функций: **nc_tag_drop(\$cc, \$message)** и **nc_tag_add(\$sub, \$cc, \$message, \$f_Tags)**, тем самым информация в базе данных тегов будет обновлена.

- В действии после удаления следует прописать вызов функции **nc_tag_drop(\$cc, \$message)**, подробное описание и список параметров будет приведён ниже, более развёрнуто.

- Для возврата к списку объектов, следует написать следующий код:

```
if ($inside_admin) {
    ob_end_clean();
    header("Location: $goBackLink&inside_admin=1");
    exit;
} else {
    echo NETCAT_MODERATION_MSG_OBJADD;
    echo "<br><br>".$goBack;
}
```

Для использования компонента «Теги» в системных настройках нашего компонента нужно добавить строку следующего содержания:

```
if($tags_messages) $query_where = "(a.Message_ID IN (".$tags_messages."))";
```

Это позволит фильтровать выводимые сообщения по прикреплённым к ним тегам.

Если требуется вывести список тегов сообщения в отображении объекта в списке, следует внести туда строку следующего содержания:

```
".listQuery("SELECT a.Tag_ID, b.Tag_Text FROM Tags_Message AS a LEFT JOIN
Tags_Data AS b ON a.Tag_ID=b.Tag_ID WHERE Sub_Class_ID=$cc AND
Message_ID=$f_RowID GROUP BY a.Tag_ID", "<a href='/tags/?tag=\
$data[Tag_ID]&tagcc=$cc'>\$data[Tag_Text]</a> ")."
```

Вывод облака тегов

Облако тегов можно вывести относительно компонента раздела, относительно раздела и всего сайта. Облако тегов относительно раздела выведет информацию, относящуюся только к конкретному разделу, а облако тегов относительно сайта выведет информацию со всего сайта.

Стоит предположить, что оптимальным решением будет первоочерёдно проверять доступность облака тегов для компонента в разделе и, если такового не окажется, проверить доступность облака тегов для раздела, в противном случае вывести облако тегов всего сайта. Для этих целей в модуле облако тегов предусмотрено три функции вывода облака тегов, код вывода может выглядеть так:

```

<!--TAGS-->
    ".(!$tagsCloud ? opt($tagsCloud = nc_tag_cloud_sub_class($tagcc ?
$tagcc : $cc, "<a href='/tags/%TAG_LINK%TAG_SUB_LINK' style='font-size:
%TAG_HEIGHT%; \".opt(%TAG_HEIGHT>160, 'line-height:1em;').\"'>
%TAG_NAME</a>"), $tagsCloud) : "")."
    ".(!$tagsCloud ? opt($tagsCloud = nc_tag_cloud_subdivision($tagsub ?
$tagsub : $sub, "<a href='/tags/%TAG_LINK%TAG_SUB_LINK' style='font-size:
%TAG_HEIGHT%; \".opt(%TAG_HEIGHT>160, 'line-height:1em;').\"'>
%TAG_NAME</a>"), $tagsCloud) : "")."
    ".(!$tagsCloud ? opt($tagsCloud = nc_tag_cloud_all($catalogue, "<a
href='/tags/%TAG_LINK%TAG_SUB_LINK' style='font-size:%TAG_HEIGHT
%; \".opt(%TAG_HEIGHT>160, 'line-height:1em;').\"'>%TAG_NAME</a>"),
$tagsCloud) : "")."
<!--/TAGS-->

```

Подробнее эти три функции будут рассмотрены ниже.

Функции модуля

● **nc_tag_add(\$sub, \$cc, \$msgID, \$f_Tags)** – функция добавления в базу информации о тегах. **\$sub** – идентификатор раздела; **\$cc** – идентификатор компонента в разделе; **\$msgID** – идентификатор самого сообщения; **\$f_Tags** – имя поля, в котором содержатся теги, перечисленные через запятую.

● **nc_tag_drop(\$cc, \$message)** – функция удаления из базы данных информации о сообщении, которое находится в определённом компоненте в разделе. Автоматически будет обработана информация о связи тегов с этим сообщением.

● **nc_tag_cloud_all(\$site_ID="", \$design, \$quantity="")** - данная функция выводит облако тегов со всего сайта, обратите внимание на необязательный параметр **\$site_ID**, он указывает на идентификатор сайта, будет полезен, когда в системе установлено более одного сайта; **\$quantity** – необязательный параметр, указывающий, сколько тегов выводить в облаке, если он не указан, то будет учитываться значение, заданное в настройках модуля; **\$design*** – содержит информацию о визуальном представлении каждого выводимого тега, может выглядеть следующим образом:

```

"<a href='/tags/%TAG_LINK%TAG_SUB_LINK' style='font-size:%TAG_HEIGHT%;
\".opt(%TAG_HEIGHT>160, 'line-height:1em;').\"'>%TAG_NAME</a>"

```

где присутствуют псевдопеременные:

● **%TAG_LINK** – ссылка формата ?tag=N, а N – идентификатор тега;

●**%TAG_SUB_LINK** – ссылка, которая будет указывать, откуда выводится тег, например &tagcc=M, M в данном случае идентификатор компонента в разделе. Это нужно для того, чтобы продолжать просматривать теги из текущего компонента, в противном случае теги будут всегда выводиться со всего сайта;

●**%TAG_HEIGHT** – высота шрифта в процентном соотношении, частовстречаемые теги будут больше;

●**%TAG_NAME** – сам тег, возможно как латинское, так и кириллическое написание;

* - аналогично для функций nc_tag_cloud_subdivision() и nc_tag_cloud_sub_class().

●**nc_tag_cloud_subdivision(\$Sub_ID, \$design, \$quantity="")** - функция вывода облака тегов из определённого раздела и всех вложенных в него разделов, в которых есть теги. **\$Sub_ID** – идентификатор раздела; **\$design** – аналогично функции nc_tag_cloud_all(), подробное описание выше; **\$quantity** – необязательный параметр, указывающий, сколько тегов выводить в облаке, если он не указан, то будет учитываться значение, заданное в настройках модуля;

●**nc_tag_cloud_sub_class(\$Sub_Class_ID, \$design, \$quantity="")** - функция вывода облака тегов из определённого компонента в разделе. **\$Sub_Class_ID** - идентификатор компонента; **\$design** – аналогично функции nc_tag_cloud_all(), подробное описание выше; **\$quantity** – необязательный параметр, указывающий, сколько тегов выводить в облаке, если он не указан, то будет учитываться значение, заданное в настройках модуля;

Вывод результатов при выборе тега и компонент «Теги»

Для вывода результатов выборки по тегам предусмотрен специальный компонент «Теги». По сути, этот компонент выводит информацию из других разделов при помощи функции s_list_class() (Руководство разработчика - Приложение 2).

При выборе тега из облака мы попадаем на скрытый раздел «Теги» в котором установлен данный компонент.

Если тег был выбран из облака относительно раздела, то в ссылку будет добавлена переменная **\$tagsub**, которая будет означать, что результаты выборки по тегу следует осуществлять только с учётом данного раздела. Например http://мой_сайт.py/tags/?tag=56&tagsub=245. Похожая переменная есть при выводе облака относительно компонента в разделе - **\$tagcc**. Эта переменная содержит идентификатор компонента в разделе. В шаблонах вывода тегов эти переменные содержатся в псевдопеременной **%TAG_SUB_LINK** (см. функции вывода облака тегов).

Календарь

Модуль предназначен для вывода датированного архива публикаций, статей, новостей и пр. объектов в виде календаря. Основанный на технологии AJAX, данный модуль позволяет производить поиск по датам сообщений без перезагрузки основного содержимого страницы. Есть возможность создавать и выводить различные структурные и цветовые темы отображения календаря.

Практическая польза и применение

Календарь поможет отфильтровать многочисленные данные по определённым датам, что существенно облегчит усилия и сократит время, потраченное на поиск информации со стороны пользователя. Чаще всего имеет смысл разместить календарь в определённом месте макета дизайна, как правило, это правая верхняя сторона структурной сетки сайта. Но конкретной привязки к местоположению данного элемента на сайте нет.

Технические требования

Включенная поддержка технологии JavaScript со стороны браузера клиента.

Требования к компонентам

Компонент вывода сообщений должен содержать поле **Date**, тип поля «Дата и время», в формате **event** отмеченное как обязательное для заполнения. В противном случае, календарь не будет учитывать данные компонента для вывода календаря, тем не менее, информация будет выведена без сформированных ссылок. При наличии в выводимых данных сообщений с заполненным полем **Date**, календарь автоматически сформирует ссылку формата **http://мой_сайт.py/.../гггг/мм/дд/** в ячейке отображения даты.

Кроме того, ссылку на публикации можно получить без уточнения дня и месяца, в этом случае фильтрация сообщения будет осуществляться без привязки к этим полям, например ссылка **http://мой_сайт.py/news/2007/07/25/** приведёт пользователя на страницу новостей за определённое число известного месяца и года, в то время, как «урезанная» ссылка **http://мой_сайт.py/news/2007/07/** отобразит сообщения за определённый месяц текущего года.

Функции модуля

● **nc_set_calendar(\$theme)** – функция инициализации данных для отображения календаря, задаётся в макете дизайна, располагается внутри тега `<head>...</head>*`, например:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01
Transitional//EN">
<html>
<head>
  <meta http-equiv='content-type' content='text/html;
  charset=windows-1251'>
  <style type='text/css'>%CSS</style>
  <title>Мой сайт</title>
  ".nc_set_calendar()."
</head>
<body>
...
</body>
</html>
```

Необязательный параметр `$theme` - идентификатор цвето-структурной схемы отображения календаря, задаётся в панели управления модуля внутри административной зоны сайта, далее будет описано более подробно.

● **nc_show_calendar(\$theme = 0, \$cc = 0, \$setDate = 0, \$DateField = "Date", \$filled = false)** - функция указывает на местоположение блока с календарём в структуре сайта. Например:

```
<table>
<tr valign='top'>
  <td width='80%'>...</td>
  <td width='20%'>".nc_show_calendar()."</td>
</tr>
</table>
```

В функции присутствует ряд необязательных параметров. `$theme` - идентификатор цвето-структурной схемы отображения календаря, если в функции `nc_set_calendar()` указан параметр темы, то и в этой функции следует указать этот параметр.

Параметр `$cc` указывает, из какого компонента в разделе забирать данные для вывода. По умолчанию берётся глобальное значение.

Параметр `$setDate` указывает дату «по умолчанию» при выводе календаря.

Параметр `$DateField` указывает, из какого поля производить выборку, по умолчанию это поле «Date». Чтобы обеспечить максимальную совместимость, на сайте следует использовать одно название для этого поля, например тот же «Date».

Параметр `$filled` указывает на «заливку» всех дней в календаре, не зависимо от значения поля вывода. В таком случае все дни будут подсвечены ссылками.

● **`nc_calendar_possibility_check($classID, $DateField="Date")`** - функция проверки наличия поля `$DateField` в компоненте `$classID`. Например:

```
<table>
<tr valign='top'>
  <td width='80%'>...</td>
  <td width='20%'>".
  (nc_calendar_possibility_check($classID, "Date") ?
  nc_show_calendar() : "Нет календаря")."</td>
</tr>
</table>
```

Панель управления модулем

В панели управления модулем реализован функционал, достаточный для внесения изменений как в цветовую схему календаря, так и для реализации структурных изменений. Кроме того, есть возможность создания новой схемы с нуля. Структурную составляющую можно реализовать в различных вариациях, как с помощью табличной разметки, так и с использованием семантической верстки. Настраиваемая таблица стилей CSS для календаря облегчает эту задачу.

Для удобства работы со структурными данными в панели управления модулем присутствует цветовая легенда, помогающая при создании и изменении схем календаря. Также, не покидая административную зону, можно просмотреть результат изменений во всплывающем рорир-окне.

В выпадающем списке со схемами календаря прописаны идентификаторы схем, их стоит использовать в функции инициализации **`nc_set_calendar($id)`**, если нужно жестко указать используемую цветовую схему. Если идентификатор отсутствует, будет выведена тема «по умолчанию», которую можно установить в панели управления модулем.

Как правило, затруднений в работе с данной панелью управления не возникает, поэтому процедура редактирования или создания новых схем для этого модуля достаточно проста. Наличие некоторого количества предустановленных схем дает более интуитивное представление о работе с модулем.

Блог и сообщество

Модуль «Блог и сообщество» позволяет создавать блоги и сообщества на Вашем сайте.

Начиная с версии системы 3.5, в модуле добавлен функционал комментариев на основе модуля "Комментарии". При создании нового корневого раздела с блогами из административной панели управления модулем, функционал "старых" комментариев по умолчанию выключен.

Действия "добавление", "редактирование" и "удаление" сообщений работают через обновлённый функционал системы. В компонент "записи блога" можно добавить поле типа "файл".

Требования к компонентам

Для работы с модулем предусмотрено пять компонентов, объединённых в единую группу «Блог». Рассмотрим каждый из них.

- **Список блогов** – титульная страница блога. Основное назначение – выводить последние добавленные сообщения, список блогов и прочую полезную информацию. Для изменения внешнего вида и функциональности достаточно изменить компонент, но не забывайте делать резервные копии компонентов или создавать их на основе существующих, без обновления последних.
- **Записи блога** – основной компонент, отвечающий за вывод сообщений внутри определённого блога. Принцип работы аналогичен всем классическим компонентам.
- **Настройки блога** – компонент, связывающий back-office с front-office. Позволяет изменять настройки разделов, к которым он прикреплен.
- **Пользователи блога** – этот компонент отвечает за управление пользователями блога. В различных типах блога (два основных) позволяет изменять доступ для пользователей, заносить в чёрный список, удалять и т.д.
- **Комментарии блога** – компонент позволяет прикреплять комментарии к сообщениям блога.

Стоит заметить, что все эти компоненты подключаются модулем автоматически при создании какого-либо типа блога. Т.е. подключать их вручную и устанавливать нужные права нет никакого смысла.

Функции модуля

Несмотря на большое количество функций, используемых для формирования блогов, всё, что требуется, можно настроить в шаблонах, которые автоматически распределяются системой при создании новых блогов. Ниже перечислены основные функции, которые могут понадобиться для расширения функциональности блогов:

- **nc_blog_show_blogs()** – функция вывода листинга блогов, устарела, вместо неё используется функция **nc_blog_list()**.

● **nc_blog_list(\$sub, \$template = "", \$orderBy = "", \$ignore_access = "")** - Функция возвращает форматированный список блогов из раздела \$sub сколько блогов, сами блоги, ссылки на них, сколько сообщений и юзеров.

\$sub - идентификатор директории с блогами

\$template - шаблон вывода списка, аналогичный шаблонам вывода навигации.

\$orderBy - порядок сортировки списка, например "Users DESC, Subdivision_Name ASC"

\$ignore_access – необязательный параметр, указывает на то, какой уровень доступа должен быть игнорирован при формировании результата, задаётся в виде числа или массива чисел, основывается на идентификаторах доступа, заданных в списке **BlogMessagesAccess**.

● **nc_blog_get_blogs(\$sub, \$orderBy = "", \$ignore_access="")** -

Функция возвращает массив с полными данными о разделе с блогами, сколько блогов, сами блоги, ссылки на них, сколько сообщений и юзеров. Используется как вспомогательная функция для функции

nc_blog_list(). **\$sub** - идентификатор директории с блогами

\$orderBy - порядок сортировки списка, например "Users DESC, Subdivision_Name ASC", поля сортировки - ключи результирующего массива

\$ignore_access - что игнорировать

Возвращает массив, каждый элемент которого - блог в разделе **\$sub**.

●**nc_blog_check_name(\$sub, \$blogname)** – функция проверяет, свободно ли имя **\$blogname** в разделе **\$sub**. Если блог (раздел) с таким именем уже есть в этом разделе, функция возвратит его идентификатор.

Например:

```
".(nc_blog_check_name(123, "techno") ? "такой блог уже существует" : "")."
```

●**nc_blog_user_permission(\$sub, \$cc)** – функция возвратит права данного пользователя на текущий раздел или компонент. Формат возвращаемого значения – «u», «m», «a». Что соответственно означает – пользователь, модератор, администратор.

●**nc_blog_get_type(\$sub, \$is_child=false)** – функция возвращает тип блога. **\$sub** – идентификатор раздела. Если раздел является блогом, то вызов функции должен осуществляться с параметром **\$is_child=1**. Если раздел **\$sub** является корневой директорией для блогов (корнем блогов), то дополнительный параметр не требуется.

Например, для корневой директории:

```
".opt($blog_type=nc_blog_get_type(45), "")."
```

или для блога пользователя, который находится в корневой директории:

```
".opt($blog_type=nc_blog_get_type(55, 1), "")."
```

●**nc_blog_get_creator(\$sub)** – функция возвращает создателя блога с идентификатором **\$sub**. **\$sub** – идентификатор раздела блога.

Например:

```
".opt($blog_creator=nc_blog_get_creator(55), "")."
```

●**nc_blog_get_blogtype()** – функция является устаревшим аналогом **nc_blog_get_type()**. В данный момент следует использовать новую функцию **nc_blog_get_type()**.

●**nc_blog_visual_settings(\$sub, \$VS_template_Header="", \$VS_template_Object="", \$VS_template_Footer="")** – функция вывода

визуальных настроек во front-office. **\$sub** – идентификатор раздела, в котором осуществляется настройка; **\$VS_template_Header** - шапка макета вывода; **\$VS_template_Object** – объект в макете; **\$VS_template_Footer** – футер макета вывода. Последние 3 параметра отвечают за визуальное оформление настроек, в частности, в них можно передать таблицу, разделённую на три части. Предполагается вызов этой функции внутри формы добавления:

```
<form name='adminForm' enctype='multipart/form-data'
method='post' action='/netcat/modules/blog/add.php'>
  <input name='sub' value='$sub' type='hidden'>
  <input name='cc' value='$cc' type='hidden'>
  <input name='nc_blog_operation' value='visual_settings'
type='hidden'>
  ".NC_Blog_Visual_Settings($sub, $VS_template_Header,
$VS_template_Object, $VS_template_Footer)."
  <input value='Сохранить' type='submit'>
</form>
```

● **nc_blog_is_friend(\$friendID)** – данная функция определяет, является ли пользователь с идентификатором **\$friendID** другом текущего пользователя. Предполагается применение только для личных блогов.

● **nc_blog_bbcode()** – функция в данный момент не используется, т.к. Её функционал входит в состав функции **nc_bbcode()**.

● **nc_blog_cut()** - функция в данный момент не используется, т.к. Её функционал входит в состав функции **nc_bbcode()**.

Панель управления модулем

В панели управления предусмотрена функция «полуавтоматического» создания раздела блогов или раздела сообществ. Следует лишь выбрать каталог, в котором предполагается создать раздел с блогами, указать имя раздела (маленькие латинские буквы, цифры, знак подчеркивания или дефис), его название в карте сайта и выбрать опции подключения RSS, подключения функционала комментариев, активность раздела. Опции RSS и комментарии будут активны для каждого впоследствии созданного блога.

После сохранения раздел блогов автоматически присоединится к заданному сайту со всеми нужными компонентами.

Настройки модуля

- **BLOG_CLASS_ID** – идентификатор компонента «Корень блога»
- **BLOG_MESSAGES_CLASS_ID** – идентификатор компонента «Записи блога»
- **BLOG_MESSAGES_SETTINGS_CLASS_ID** – идентификатор компонента «Настройки блога»
- **BLOG_MESSAGES_USERS_CLASS_ID** – идентификатор компонента «Пользователи блога»
- **BLOG_COMMENTS_CLASS_ID** – идентификатор компонента «Комментарии блога»
- **BLOG_RSS_TEMPLATE_ID** – идентификатор макета «RSS для блогов»
- **BLOG_USER_PERMISSION** – права пользователей для таблицы Permission
- **BLOG_MODERATOR_PERMISSION** – права модераторов для таблицы Permission
- **BLOG_ADMIN_PERMISSION** – права администраторов для таблицы Permission
- **BLOG_NOTICE_BLOCK** – оформление блока вывода системных сообщений и предупреждений

Управление действиями в компонентах

Для предоставления максимального доступа к изменению внешнего вида компонентов для большинства действий (добавление пользователя в друзья, сохранение настроек, присоединение к сообществу...) используются формы. Ключевым параметром, определяющим текущее действие, является переменная **nc_blog_operation**. В зависимости от значения данной переменной, система выбирает нужное действие. Например, добавление пользователя в друзья:

```
<form enctype='multipart/form-data' method='post'
action='/netcat/modules/blog/add.php'>
  <input name='sub' value='$sub' type='hidden'>
  <input name='cc' value='$cc' type='hidden'>
  <input name='nc_blog_operation' value='add_user' type='hidden'>
  <input value='Добавить в друзья' type='submit'>
</form>
```

Обратите внимание, что параметр action должен быть следующим:

`action='/netcat/modules/blog/add.php'`

NetCat 4.5 Руководство по модулям

Список доступных действий:

- **create_blog** – создать новый блог. Дополнительное поле для создания блога внутри сообщества `<input type='text' name='blogName'>` определяет ключевое слово для блога.
- **join_blog** – присоединиться к блогу. Актуально для сообществ.
- **unjoin_blog** – прекратить участие в сообществе. Актуально для сообществ.
- **add_user** – добавление пользователя, в блоге которого находится текущий пользователь, в друзья к последнему. Актуально для личных блогов. Если присутствует параметр **ignore_user**, заданный посредством поля `<input name='ignore_user' value='...' type='hidden'>`, то пользователь будет добавлен в список игнорируемых (текущий пользователь находится в своём блоге).
- **drop_user** – действие обратное **add_user**.
- **save_users** – сохранить/обновить список пользователей. Актуально для всех типов блогов, предполагается использовать в компоненте «Пользователи блога»
- **visual_settings** – сохранить/обновить настройки блога. Актуально для всех типов блогов, предполагается использовать в компоненте «Настройки блога». Сами настройки задаются в поле «Настройки отображения компонента раздела» компонента «Записи блога».

После сохранения раздел блогов автоматически присоединится к заданному сайту со всеми нужными компонентами.

Защита форм картинкой

Модуль «Защита форм картинкой» (CAPTCHA, Completely Automated Public Turing test to tell Computers and Humans Apart) позволяет защищать формы изображением, которое содержит набор символов, легко читаемые людьми, но с трудом поддающиеся распознаванию автоматами. Таким образом, CAPTCHA — это дополнительное препятствие для автоматизированных средств размещения информации на сайтах (форумах, гостевых книгах, формах регистрации и обратной связи, и т.п.)

В составе системы NetCat модуль поставляется, начиная с версии 2.4.3.

Для функционирования модуля необходима установленная библиотека для работы с графическими изображениями GD2.

Принцип работы модуля

Чтобы защитить форму с помощью модуля Captcha, в ней должны присутствовать изображение, на котором находится код подтверждения, скрытое поле, через которое передается хэш-код, ассоциированный с кодом подтверждения и поле для ввода пользователем кода подтверждения.

Хэш-код генерируется либо напрямую вызовом функции `nc_captcha_generate_hash()`, либо косвенно вызовом `nc_captcha_generate_formfield()`. Хэш-код хранится в базе данных и по нему происходит проверка корректности кода подтверждения.

На странице валидации формы происходит вызов функции `nc_captcha_verify_code()`, которая сравнивает введенный пользователем код подтверждения и значение в базе данных, индексированное хэш-кодом.

Настройка модуля

Настройки модуля «Защита форм картинкой» доступны через интерфейс управления модулями в системе управления NetCat. Путь к настройкам: Инструменты — Модули — Настройки модуля «Защита форм картинкой». В первоначальной поставке модуля доступны следующие параметры:

ALPHABET=ABCEFGHKLMNPRSTUVWXYZ

Набор символов, из которых складывается код подтверждения, отображаемый на картинке. Можно использовать латинские символы и арабские цифры. Кириллические символы отображаются неверно, по причине их недоступности в библиотеке GD2, с помощью которой формируется изображение.

NUMBER_OF_CHARS=3..6

Количество символов в коде подтверждения. Если параметр указан в виде "X..Y", то код будет состоять из случайного количества символов (от X до Y символов). Если указано одно число, код будет всегда состоять из данного числа символов.

DURATION=1800

Срок хранения информации о сгенерированном коде, в секундах. Первоначальное значение — 30 минут ($1800 = 30 * 60$). По истечении срока хранения, сгенерированный код становится в любом случае неверным. Если вы защищаете большую форму, на заполнение которой пользователю потребуется много времени, увеличьте значение этого параметра.

IMAGE_WIDTH=150

Ширина генерируемой картинки, в пикселях.

IMAGE_HEIGHT=30

Высота генерируемой картинки, в пикселях.

NUMBER_OF_LINES=15

Дополнительная защита картинки от автоматического распознавания размещает на фоне произвольно проведенные прямые. Данный параметр отвечает за их количество.

HIDDEN_FIELD_NAME=nc_captcha_hash

Имя скрытого поля в форме, через которое передается хэш-код, соответствующий показанному пользователю изображению.

AUDIOCAPTCHA_ENABLED = 0

Включить аудиокапчу

Вы можете добавлять собственные параметры, которые будут использоваться в модифицированной версии генерации картинки, см. «Изменение функции генерации картинки».

Защита стандартных форм

Для того, чтобы добавить картинку к форме добавления объекта, установите флаг "Защищать форму добавления картинкой" в настройках шаблона данных или шаблона (если вы хотите, чтобы картинка показывалась в форме добавления во всех разделах) или в настройках шаблона конкретном разделе. Картинка с кодом будет показана в формах добавления только для тех пользователей, которые не вошли на сайт (не зарегистрированы в системе).

Защита формы регистрации пользователей

Для добавления картинки с кодом для формы регистрации пользователей установите флаг "Защищать форму добавления картинкой" в настройках системной таблицы "Пользователи".

Защита формы добавления в модуле "Форум"

Для защиты формы добавления сообщения на форуме необходимо отредактировать соответствующие шаблоны. Для этого откройте список шаблонов в разделе "Форум" и отредактируйте объекты в соответствующих шаблонах ("Форум: редактирование сообщения" и "Форум: ответ на сообщение").

В шаблоне вывода формы добавления сообщения необходимо добавить блок, ограниченный тэгами `<!--%Captcha_block_begins%-->` и `<!--%Captcha_block_ends%-->`. Данный блок не будет показываться пользователям, которые вошли в систему.

Следующие макропеременные будут заменены на соответствующие значения:

- `-%Captcha_image%` - картинка с кодом и скрытое поле с хеш-кодом (обязательно должно присутствовать в блоке);
- `-%Captcha_message%` - сообщение "Введите символы, изображенные на картинке" (Вы можете ввести вместо этой переменной любой текст);
- `-%Captcha_field%` - поле для ввода кода (обязательно должно присутствовать в блоке).

```
<!--%Captcha_block_begins%-->
```

```

<tr>
  <td>&nbsp;</td>
  <td colspan="4">
    <br>
    %Captcha_image%<br><br>
    %Captcha_message%: %Captcha_field%<br><br>
  </td>
</tr>
<!--%Captcha_block_ends%-->

```

Если Вы не хотите, чтобы форма добавления сообщения на форум была защищена картинкой, удалите блок `<!--%Captcha_block_begins%-->...<!--%Captcha_block_ends%-->` из шаблонов "Форум: редактирование сообщения" и "Форум: ответ на сообщение".

Для отображения сообщения о возникновении ошибки используйте `%Captcha_wrong_code%`, при этом будет выведено стандартное сообщение «Неправильно введено число на картинке». Изменить текст сообщения нельзя.

Защита альтернативных форм

В альтернативной форме добавления необходимо поместить код:

```

".nc_captcha_formfield()."
Введите код на картинке: <input type='text'
name='nc_captcha_code' value=''>

```

В поле "Условия добавления":

```

if ( !nc_captcha_verify_code($nc_captcha_code,
$nc_captcha_hash) ) {
  $posting = 0;
  $warnText = "Неправильный код!";
}

```

Функции, доступные после установки модуля

После установки модуля Captcha в макетах дизайна и в компонентах доступны следующие функции:

```
nc_captcha_formfield
```

```
string nc_captcha_formfield ( [string $attributes],  
[string $beg], [string $mid], [string $end] )
```

Возвращает строку, которая содержит поле типа `hidden`, хранящее хэш-код от сгенерированной картинки и кода подтверждения на ней и `html`-код картинки с кодом подтверждения.

Хэш-код генерируется вызовом функции `nc_captcha_generate_hash`. Возвращает готовую разметку, пригодную для вставки в форму. Функция принимает необязательный аргумент с дополнительными атрибутами для тэга `img`. Примеры использования:

```
nc_captcha_formfield()  
    возвращает строку вида:  
    <input type='hidden' name='nc_captcha_hash'  
    value='d41d8cd98f00b204e9800998ecf8427e'><img  
    src='/netcat/modules/captcha/img.php?  
    code=d41d8cd98f00b204e9800998ecf8427e'>  
nc_captcha_formfield('width="150" height="30" border="0"')  
    возвращает строку вида:  
    <input type='hidden' name='nc_captcha_hash'  
    value='d41d8cd98f00b204e9800998ecf8427e'><img  
    src='/netcat/modules/captcha/img.php?  
    code=d41d8cd98f00b204e9800998ecf8427e' width="150"  
    height="30" border="0">
```

\$beg, \$mid, \$end — отвечают за отображения аудио капчи.

Значения по умолчанию:

\$beg:

```
<span class='nc_captcha_voice' style='margin:0 0 0 10px;'><a  
class='nc_captcha_js' href='#' onclick='nc_captcha.play();return false;'>&#9834;
```

\$mid:

`NETCAT_MODULE_CAPTCHA_AUDIO_LISTEN` — макропеременная, содержащая «Прослушать».

\$end:

```
</a><span id='nc_captcha_player'  
style='display:none;'></span></span>
```

nc_captcha_generate_hash

```
string nc_captcha_generate_hash ()
```

При вызове функция генерирует код подтверждения на основе настроек модуля и уникальный хэш-код для него (хэш-код не зависит от кода подтверждения), после чего сохраняет их в базе данных. В дальнейшем этот хэш используется для проверки введенного пользователем кода подтверждения. Хэш-код действителен в течении времени, указанного в настройках модуля (DURATION). При каждом вызове функции генерируется новый код подтверждения и новый хэш-код для него.

Функция не принимает аргументов. Возвращает хэш-код.

Код следует передавать через поле типа hidden на страницу валидации формы (обычно это Условие добавления объекта). На основе хэш-кода также генерируется картинка с кодом подтверждения. Путь к картинке: /netcat/modules/captcha/img.php?

code=d41d8cd98f00b204e9800998ecf8427e. Пример использования в альтернативной форме добавления объекта:

```
".opt($captcha_hash = nc_captcha_generate_hash(), "")."  
<input type='hidden' name='nc_captcha_hash'  
value='$captcha_hash'>  
<img src='/netcat/modules/captcha/img.php?  
code=$captcha_hash'><br>  
Введите код на картинке: <input type='text'  
name='nc_captcha_code' value=''>
```

Примечание: в большинстве случаев достаточно использовать функцию nc_captcha_formfield; пример использования приведен выше.

nc_captcha_verify_code

```
bool nc_captcha_verify_code ( string $user_code [,  
string $user_hash] [, bool $delete_hash = 1] )
```

Проводит проверку корректности введенного пользователем кода подтверждения и полученного из этой же формы хэш-кода. Функция принимает от одного до двух аргументов. Первый – код подтверждения, введенный пользователем. Второй – необязательный хэш-код, по которому следует устанавливать корректность кода подтверждения. Если второй аргумент опущен, функция пытается получить его из массивов \$_GET и \$_POST (по имени, заданному в параметре HIDDEN_FIELD_NAME

настроек модуля). Функция учитывает время действия (параметр DURATION в настройках модуля) хэш-кода.

Возвращает true, либо false. Пример использования в условии добавления объекта:

```
Работает с предыдущим примером
if ( !nc_captcha_verify_code($nc_captcha_code,
$nc_captcha_hash) ) {
    $posting = 0;
    $warnText = "Здравствуй, робот!";
}
```

Если параметр \$delete_hash будет равен false, то хэш удаляться не будет.
nc_captcha_image

```
string nc_captcha_generate_image ( string
$captcha_hash )
```

Функция содержит алгоритм генерации картинки с заданной последовательностью символов на основе хэш-кода. Возвращает картинку в формате gif. Подробнее см. Модификация модуля Captcha.

nc_captcha_generate_code

```
string nc_captcha_generate_code ( string $captcha_hash
)
```

Функция создает случайный код и сопоставляет его хэшу, указанному в параметре \$captcha_hash. Длина кода и набор символов, из которых состоит код, задается в настройках модуля.

Во всех примерах использован хэш-код d41d8cd98f00b204e9800998ecf8427e, который является результатом выполнения функции md5 от пустой строки.

Модификация модуля Captcha

Возможна замена стандартного алгоритма генерации картинок на пользовательский. Для этого в файле /modules/captcha/user_functions.inc.php добавьте функцию nc_captcha_user_image(); в качестве первого параметра ей будет передаваться код, который должен быть изображен на картинке.

Функция должна возвращать строку, которую можно отдать браузеру в качестве картинки (gif, jpeg, png).

Аудио каптча

В версии NetCat 4.5 появилась возможность использовать аудио капчу. Для её активации достаточно пройти в настройки модули и установить переменную `AUDIOCAPTCHA_ENABLED` в **1** (единицу).

Переменная `VOICE` задает голос аудио капчи, файлы для неё лежат по адресу `/netcat/modules/captcha/voice/`. Вместе с системой поставляется голос «Natasha». Подготовив и залив голосовые файлы в необходимом формате, вы сможете добавлять свои голоса.

Модуль "Кэширование"

Описание модуля

Предисловие

Кэш (англ.cache) — *"это промежуточный буфер с быстрым доступом, содержащий копию той информации, которая хранится в памяти с менее быстрым доступом, но с наибольшей вероятностью может быть оттуда запрошена. Доступ к данным в кэше идёт быстрее, чем выборка исходных данных из медленной памяти или их перевычисление, что делает среднее время доступа короче"*.

В случае, применимом к специфике работы системы управления сайтами NetCat, кэш содержит в себе данные, полученные путём выполнения запросов к базе данных и обработки их различными функциями. Обработанные данные хранятся на дисковом пространстве сервера и могут быть получены путём чтения их из файла. При сложных многоуровневых запросах к базе данных и их обработке без использования кэширования, затрачивается время много большее, (с точки зрения системы и программного интерпретатора), чем несложный функционал проверки доступности кэшированного результата и чтение его из файла кэша.

Существует множество разновидностей кэширования каких бы то ни было данных — кэширование результатов выполнения функций, кэширование запросов к базе данных, кэширование большого объёма информации полученной из разных источников и выводимой в едином контексте. Кэш применяется повсеместно в среде программирования, обработке данных и сложных вычислений, более того, многие программные продукты немыслимы без такого функционала, особенно если речь идёт о серьёзных ресурсоёмких проектах.

Конечно, кэширование — не панацея, но в умелых руках этот инструмент позволит значительно сократить нагрузки на сервер, тем самым облегчив жизнь программисту, системному администратору хостинга и, конечно же, пользователю.

Примеры из жизни

Представьте несложную страничку содержащую ленту новостей, анонс последних, наиболее значимых, событий в жизни компании и информацию о вакансиях, отсортированных, например, по уровню заработной платы. Казалось бы, всё просто и прозрачно, но, естественно, должно быть какое-то "но". Новости хранятся в таблице новостей, данные об авторе в таблице пользователей, события — в таблице мероприятий, фотоматериалы к тексту — в таблице соответствия файлов, да и вакансии стоит получить неким запросом к базе данных, отсортировав в нужном порядке. Даже в этом примитивном примере собирается неплохая подборка запросов на вывод одной страницы. А пользователь будет не один, и такие идентичные операции будут повторяться раз за разом для каждого посетителя сайта. Ресурсы сервера не резиновые, повышение количества запросов приведёт к потере производительности во времени.

Естественно если сайт небольшой и количество обращений к нему не так уж велико, можно не заметить существенного снижения скорости отдачи страниц с результатом. Но чем популярнее будет становиться сайт и чем больше пользователей захотят его посетить — тем большее время будет затрачивать сервер на генерацию каждой страницы. При продолжающемся росте нагрузки это может привести к критическим последствиям — сервер перестанет справляться с нагрузкой и сайт станет недоступен пользователям.

Принцип работы кэша

Как же работает кэш? Всё предельно просто: когда пользователь открывает страницу, система проверяет, есть ли уже обработанная копия результата для запрашиваемой страницы на диске. Если такой копии нет, система создаст страницу "на лету", в обычном режиме. Перед тем как отправиться пользователю, страница будет сохранена на диске. Когда такую же страницу запросит другой пользователь, система проверит наличие обработанного результата, получит положительный отзыв, и, вместо повторной генерации и многократных запросов к базе, отдаст пользователю сохраненное содержимое из файла кэша.

Файлы кэша расположены в папках на сервере, связанных с определёнными сущностями (сайт, раздел и компонент в разделе). Имя файла строится на основе адреса запрашиваемой страницы. Стоит иметь ввиду, что страница с

адресом `/news/news_5.html` и страница `/news/news_5.html?curPos=2` для системы — два разных набора данных для кэширования, которые будут храниться в разных файлах, но в одной папке на сервере.

Естественно информация на сайте может меняться и логично предположить изменение кэшированных данных. Большинство операций система берёт на себя и если информация изменится через функции редактирования, удаления или добавления системы, — устаревший файл кэша для этого объекта(ов) будет удалён.

Может возникнуть ситуация, когда кэшируемые данные будут содержать в себе динамические данные, которые так же кэшируются, но на другой странице. Получается, что на диске сервера будет создано два файла кэша и, в то же время, изменения вложенного кэша не смогут повлиять на кэш основной. В таком случае разработчик должен сам контролировать подобные взаимосвязи. Далее будет рассмотрен функционал частичного кэширования с применением конструкции `"nocache"`.

Тем не менее, очень часто нет необходимости кэшировать все данные на сайте. Например, нельзя кэшировать закрытые данные из профайла пользователя, иначе они станут доступны другим посетителям сайта. В предлагаемом модуле кэширования есть решение для данной задачи — блоки `"nocache"`. О них будет сказано чуть позже.

Рассмотрим теперь непосредственно использование кэширования в рамках системы NetCat.

Использование функционала кэширования

На данный момент есть следующие возможности кэширования:

Кэширование результатов вывода списка объектов, функция `s_list_class()` наиболее часто используемая как самой системой, так и пользователями создающими сайты на системе NetCat, поля "Префикс списка объектов", "Объект в списке" и "Суффикс списка объектов" в компоненте (раздел "Разработка" — "Список компонентов").

Кэширование результатов вывода подробного описания объекта, поле "Отображение объекта" в компоненте (раздел "Разработка" — "Список компонентов").

Кэширование результатов вывода функций навигации по сайту `s_browse_catalogue()`, `s_browse_sub()` и `s_browse_cc()` (подробное описание в руководстве разработчика "Приложение 2. Список используемых функций и переменных").

Кеширование результатов выполнения функций. Расширяемый функционал.

Кэширование вывода календаря (модуль "Календарь"). Функция `nc_show_calendar()`.

Функционал кэширования может быть включен в настройках каталога ("Сайт" — "Имя сайта" — "Настройки"), так же общее включение и выключение кэширование устанавливается в настройках модуля "Кэширование" (эти настройки обсуждаются далее по тексту). При установленном модуле кэширования на этой странице появляется блок с одноимённым названием. По умолчанию кэширование отключено, чтобы задействовать его, достаточно установить переключатель в положение "Разрешить", установить, если требуется, время в минутах, в течение которых кэш будет актуален (не будет удалён системой) и нажать на кнопку сохранения настроек. Всё, кэширование включено. *Создание различных видов кэша по-умолчанию включено в настройках модуля.*

Теперь рассмотрим управление кэшированием подробнее.

Кэширование наследует настройки от родительских сущностей. То есть, если включить кэширование в настройках сайта, то оно автоматически распространится на все его разделы. Если кэширование на сайте разрешено, а в каком-либо разделе этого сайта оно отключено (выведено из положения по-умолчанию "наследовать"), этот раздел кэшироваться не будет. Так же наследуется время актуальности кэша. Данная настройка, как пример установленная в положение 30 (минут), будет говорить системе о том, что через заданное время файл кэша должен быть пересоздан заново.

Аналогично каталогу, кэширование включается или выключается для разделов и компонентов в разделе (вкладка "Настройки").

Следует учесть, что кэш результатов выполнения функций контролируется только из настроек модуля, настройки различных сущностей (сайт, раздел и компонент в разделе) влияют только на возможность создания такого кэша, контроль чтения и установки актуальности кэша в данном случае не активны. Однако можно управлять

данным видом кэша из настроек модуля, об этом написано в обзоре административной части модуля.

Блок "nocache"

Если какой-либо блок информации кэшировать нежелательно, его можно обрамить следующей HTML-конструкцией:

```
<!-- nocache -->
".show_profile($user_id)."
<!-- /nocache -->
```

В таком случае система не будет включать данный блок в кэш, а прочитает его при выдаче кэша отдельно. Это не так эффективно, как кэширование всего результата полностью, но гораздо эффективнее, чем работа без системы кэширования вообще.

Стоит оговориться, что переменные `$f_имя_переменной` в блоках "nocache", так же будут доступны, в этом случае будет создан дополнительный файл кэша, с расширением `.vars.php`. На данный момент нет возможности использовать обработанные переменные `$f_имяпеременной_url` и т.п.

В таком виде блок "nocache" доступен в полях компонента "Префикс списка объектов", "Объект в списке", "Суффикс списка объектов" и "Отображение объекта" (раздел "Разработка" — "Список компонентов").

Для отмены кэширования в функциях вывода навигации, следует добавить переменную массива "nocache" равную 1 в "шаблон вывода навигации":

```
$browse['prefix'] = "<div>";
$browse['suffix'] = "</div>";
$browse['active'] = "<a href='%URL'>%NAME</a>";
$browse['active_link'] = "<a href='%URL'>%NAME</a>";
$browse['unactive'] = "<a href='%URL'>%NAME</a>";
$browse['divider'] = "";
$browse['nocache'] = 1;
```

В таком случае данный шаблон не будет кэшироваться целиком. Частично кэшировать функции вывода навигации не имеет смысла, т.к. издержки на определение и разбор блоков "nocache" перевесят полный просчёт результатов функций навигации.

Административная часть

В панели управления модуля ("Настройки" — "Список модулей" — "Кэширование") можно ввести различные опции для разных сайтов, созданных в рамках одной системы. Это позволяет гибко настраивать и изменять поведение системы кэширования для различных проектов. Кроме того, в данном разделе содержатся инструменты, позволяющие проследить эффективность работы кэша в различных сущностях системы (сайт, раздел и компонент в разделе). Рассмотрим настройки подробнее.

Настройки кэша

Сайт для настройки

Первым блоком в разделе настроек вынесен выпадающий список сайтов системы. Если сайтов несколько, следует выбрать нужный и настройки автоматически перезагрузятся. Если были внесены изменения в настройки кэша какого либо сайта — следует нажать кнопку "Сохранить", иначе настройки не будут сохранены.

Основные настройки

В этом блоке устанавливается, какой тип кэша будет активен на выбранном выше сайте.

Настройки аудита

Далее следуют настройки аудита. Рассмотрим их подробнее. В режиме аудита система кэширования помимо записи и отдачи кэша собирает информацию о проделанных действиях. В данной информации содержатся подсчёты количества попыток прочитать или записать кэш, количества прочитанного или записанного кэша, данные о сущностях связанных с кэшем (сайт, раздел и компонент в разделе; для кэша подробного отображения объекта так же записывается идентификатор компонента и сообщения). Эта информация записывается в базу данных при каждой попытке чтения или записи кэша, в последующем она поможет проанализировать эффективность кэша для разных сущностей, а также собрать информацию для автоматической очистки неэффективного кэша. Режим аудита по-умолчанию выключен, т.к. в данном режиме выполняются запросы к базе данных для сбора статистической информации.

Так как одна и та же страница с разными параметрами в строке адреса, воспринимается системой кэширования как два разных набора выводимых

данных, на основе данных аудита можно определить отдачу от кэширования в определённом разделе или компоненте в разделе.

Чтобы активировать режим аудита следует установить чекбокс "включить режим аудита" в положение "выбрано" и вписать время в часах, в течении которых режим будет активен. Датой начала режима аудита считается момент сохранения настроек. Ниже чекбокса включения аудита расположена таблица содержащая информацию о времени начала и окончания аудита, а так же поле для указания значения продолжительности режима в часах.

Настройки квот

Квоты позволяют обозначить предел дискового пространства для кэшируемой информации. Это очень важно, когда дисковое пространство ограничено, а также позволяет обезопасить файловую систему от излишней перегрузки данными. Квоты устанавливаются отдельно для разных видов кэша, их размер задаётся в мегабайтах. Как пример можно установить ограничение следующим образом:

Кэш списка объектов — 5MB

Кэш подробного отображения — 3MB

Кэш функций навигации — 1MB

Кэш результатов выполнения функций — 3MB

Кэш отображения календаря — 1MB

Если квоты не указаны или равны 0, это будет означать отсутствие ограничений по размеру.

В этом же блоке предлагается установить действия, если квоты превышены. Действие по-умолчанию — "Не кэшировать". В данном случае кэш просто не будет создаваться, пока не освободится место в папке кэша. Чтобы выбрать действие "Удалять неэффективный", нужно обладать данными в таблице очистки кэша. Эти данные можно экспортировать из данных аудита (если был включен режим аудита и таковые данные имеются в базе). Экспорт данных будет рассмотрен ниже.

Также в настройках квот можно вручную очистить всю папку определённого вида кэша. Для этого следует выделить чекбокс напротив нужной разновидности кэша и нажать на кнопку "Сохранить".

Информация

Общая информация

В этом блоке содержится информация о текущей заполненности дискового кэша, отдельно по разновидностям и итоговый.

Таблица очистки

Если в таблицу очистки неэффективного кэша были импортированы данные из таблицы аудита, система покажет количество записей и среднее число эффективности каждой из разновидностей кэша.

Внизу находятся кнопки "Удалить данные очистки", если таковые имеются, и "Обновить информацию", которая просто перезагружает текущий фрейм.

Данные аудита

Кэш

Здесь предлагается выбрать разновидность кэша (список объектов, подробное отображение и функции навигации) для отображения данных аудита, если данные были собраны. Рядом с выпадающим списком указано количество усреднённых позиций из базы данных, для соответствующего вида кэша.

Данные аудита

В зависимости от выбранного типа кэша, будут показаны собранные усреднённые данные, в которых указан сайт, раздел и компонент в разделе, а также коэффициент эффективности кэша для данной сущности. Этот коэффициент есть не что иное, как отношение количества успешных отдач кэша с диска к общему количеству попыток прочитать кэш запрашиваемой страницы. То есть, если кэш для страницы /publications/news/ запрашивался 20 раз, а был прочитан с диска только 12 раз (имеется ввиду наличие кэшированных данных), коэффициент примет значение: $12 / 20 = 0.6$, что вполне неплохо. Чем коэффициент больше — тем выше эффективность кэша, это значение так же называют *локальностью кэша*.

Система разделяет 3 степени эффективности кэша:

- от 0 до 0.2 не включительно — неэффективный кэш
- от 0.2 включительно до 0.7 — средняя эффективность, хорошая
- от 0.7 включительно до 1 включительно — отличная эффективность.

Значения этих степеней эффективности можно настраивать, подробнее об этом написано в следующем разделе.

В таблице, нажав на поле "Эффективность", можно поменять порядок сортировки отображаемых результатов.

Ниже располагаются три управляющих кнопки (их количество может зависеть от доступности разного рода информации). "Сохранить в таблице очистки" — экспорт обработанных данных в таблицу очистки, об этом упоминалось выше. "Удалить данные аудита" — очищает таблицу собранных данных в режиме аудита, данные в таблице очистке удаляются на вкладке "Информация". "Обновить информацию" — перезагрузка текущего фрейма.

Дополнительным инструментом в сборе данных аудита может служить индексация данных в модуле "Поиск по сайту" ("Настройки" — "Список модулей" — "Поиск по сайту" — "Индексировать сайт"), если он установлен в системе. Индексирующий скрипт "проходит" по разрешённым в настройках модуля адресам, тем самым пополняя таблицу данных аудита в модуле "Кэширование". Естественно кэширование должно быть включено в нужных сущностях (сайт, раздел и компонент в разделе).

Константы модуля

Важно: данные настройки предназначены только для опытных пользователей, установка значений констант отличных от значений по-умолчанию может привести к некорректной работе модуля в целом.

В административном списке предопределённых констант модуля ("Настройки" — "Настройка модуля" — "Кэширование" — "Настройки") по-умолчанию прописаны следующие обязательные константы:

● **DEBUG_MODE** — Эта константа контролирует режим включения отладки. Данный режим нужен скорее для разработчиков, чтобы отследить возникающие ошибки. В этом режиме в самом низу страницы будет выведена информация о текущих операциях с кэшем. По-умолчанию режим отключен. Чтобы включить режим, достаточно установить значение константы равной 1.

Допустимые значения: числа 0 или 1.

●**EFFICIENCY_LOW** — Константа указывает верхнее значение коэффициента эффективности "плохого" кэша не включительно, т.е. неэффективного.

Допустимые значения: целое или дробное число с 1-2 знаками после точки.

●**EFFICIENCY_MID** — Константа указывает верхнее значение коэффициента эффективности "среднего" кэша не включительно, т.е. эффективного. Стоит оговориться, что верхнее значение "хорошего" (наиболее эффективного) кэша всегда равно 1, поэтому его указывать не стоит. Диапазон коэффициента считается по выражению: $k \geq n \text{ AND } k < m$ (от n включительно, до m не включительно), где k — коэффициент эффективности.

Допустимые значения: целое или дробное число с 1-2 знаками после точки.

●**DEFAULT_LIFETIME** — Константа определяет время актуальности кэша по умолчанию. Данный параметр необходим для контроля времени жизни кэша функций, которые не могут получить данное значение из настроек каталога, раздела или компонента в разделе. Как пример — функции, которые реализуют механизм наследования.

Использование Memcached

Memcached — программа, работающая на стороне сервера и хранящая кэш в оперативной памяти. Поскольку доступ к оперативной памяти осуществляется намного быстрее, чем доступ к файлам на диске, то использование memcached может сильно снизить нагрузку на сервер и уменьшить время выдачи страницы.

Требования

Для работы с memcached'ом необходимо, чтобы на сервере был установлен и запущен сам memcached, так же должна быть установлена клиентская библиотека — расширение php под названием memcache. Так же вам необходимо знать адрес хоста memcached и порт, обычно это localhost и 11211 соответственно.

Настройка

Для включения и настройки memcached необходимо зайти в административную часть модуля, в раздел «Настройки кэша».

В области Memcached будут доступны следующие опции:

- Использовать memcached – при включении кэш будет сохраняться через сервис memcached;
- адрес сервера — адрес сервера memcached, например localhost;
- порт — порт, на котором работает сервер memcached, например 11211.

Модуль «Комментарии»

Описание модуля

Предисловие

Комментарии – необходимый атрибут современных интернет сайтов и социальных сетей. Благодаря им пользователи могут высказывать своё мнение относительно какой-либо онлайн публикации, дополнять её другой информацией, поддерживать или опровергать высказанные мнения, вести диалог с другими пользователями системы.

В системе NetCat комментарии могут быть подключены к любому объекту, добавленному в систему через компоненты. На практике это новости, записи блога, статьи, товары в интернет магазине и т. п. Комментарии обладают древовидной структурой и выводятся автоматически в виде дерева со вложенными ответами, если таковые присутствуют. Любые действия (добавление, редактирование, удаление, получение новых данных) с комментариями происходят без перезагрузки страницы, т. к. реализованы с использованием технологии AJAX.

На комментирование распространяется политика безопасности, схожая с добавлением любых данных через компоненты системы. Можно разрешить добавление комментариев всем, зарегистрированным или уполномоченным пользователям, либо вообще отключить такую возможность.

Примеры из жизни

Допустим, на научном сайте есть раздел для публикации статей на определённую тематику, например — "Новые изобретения". Автор, пользователь сайта, публикует общую информацию о новом, экологически чистом и возобновляемом топливе. Несомненно, процесс прочтения такой статьи является увлекательным занятием, но вероятно, могут возникнуть уточняющие вопросы по теоретической части. Хорошо если указан электронный адрес автора или же другая его контактная информация. Но ведь такой информации может и не быть, да и написание писем, звонки по телефону не всегда возможны, удобны и вообще желаемы. К тому же сам

автор не будет в восторге отвечать на такие запросы, потому как это требует значительного времени, да и вопросы могут повторяться раз от раза.

Гораздо удобнее вести диалог в комментариях к статье. Ко всему прочему, это избавляет от повторных вопросов и ответов. Преимущества такого способа онлайн общения весьма очевидны. Так администрация сайта или автор статьи может разрешить комментирование статей для незарегистрированных пользователей, либо наоборот отключить такую возможность.

Зачастую комментарии способны значительно расширить информацию из первоначальной публикации, что сделает прочтение комментариев весьма не бесполезным делом.

Принцип работы комментариев

Представьте, что вы прочитали какую либо статью, после которой видите ссылку "Оставить комментарий". После нажатия на эту ссылку, на экране появится форма добавления, в которую следует вписать желаемый текст и подтвердить добавление, нажатием на кнопку "Сохранить". Добавление комментария пройдет без перезагрузки страницы, после чего вы увидите свой комментарий в "стенке комментариев" (или дерево комментариев).

Следует сразу оговориться, что текст на ссылке добавления, равно как и сама ссылка, отображаются согласно шаблону вывода блоков комментирования, которые задаются в административной части системы, об этом будет написано ниже по тексту.

Иногда может понадобиться, чтобы форма для добавления нового комментария была видна сразу (без нажатия на ссылку «Оставить комментарий»). Такая возможность также предусмотрена.

В зависимости от настроек системы комментариев во вновь добавленном комментарии может присутствовать информация об его авторе, дате публикации и нескольких ссылок: "ответ на комментарий", "редактирование комментария", "удаление комментария". Сама форма добавления или редактирования может содержать панельку с ВВ-кодами.

Ответ на комментарий, по сути, является таким же комментарием, как и все остальные, за исключением того, что у него есть свой родительский комментарий, ответом на который он и является. Визуально ответы

расположены под родительским комментарием и могут содержать в себе небольшой отступ слева для усиления визуального эффекта.

Представьте, что два не связанных между собой пользователя пишут комментарий к одной и той же статье. Причём на момент написания они явно видели 3 уже добавленных комментария, оставленных другими пользователями или гостями. Пользователь №1 добавляет комментарий первым и видит свой комментарий вместе с тремя предыдущими. Через несколько секунд или минут, Пользователь №2 закончил набор текста для своего комментария, он по-прежнему видит только 3 изначальных комментария, и нажимает кнопку "Сохранить". В тот же момент его комментарий отправляется на сервер, который в свою очередь присылает ответ. В этом ответе будет содержаться информация о добавлении в "стенку комментариев" данных о комментарии Пользователя №1 и Пользователя №2. Всё это пройдёт без перезагрузки страницы. Проще говоря, система добавила комментарий Пользователя №2 и недостающий комментарий Пользователя №1. Такой же принцип взаимодействия заложен в действиях "редактирование комментария" и "удаление комментария". *Следует отметить, что обновится только та часть "стенки комментариев", которая претерпела изменения, это немаловажно при большом количестве комментариев, когда перерисовка всего дерева, а также получение больших данных средствами AJAX, может замедлить скорость работы.*

Использование функционала комментариев

Функционал комментариев может быть включен в настройках каталога ("Сайт" — "Имя сайта" — "Настройки"). При установленном модуле комментариев на этой странице появляется блок с одноимённым названием. По умолчанию комментирование включено для всех, но чтобы задействовать его, следует добавить 2-3 строки в компонент, комментирование объектов которого подразумевается. Это несложная операция, о которой будет сказано ниже в этом разделе.

Теперь рассмотрим управление комментариями подробнее.

Комментарии наследуют настройки от родительских сущностей. То есть, если включить комментирование в настройках сайта, то оно автоматически распространится на все его разделы. Если комментирование на сайте разрешено, а в каком-либо разделе этого сайта оно отключено (выведено из положения по умолчанию "наследовать"), этот раздел обретёт свои собственные правила на это действие. Так же наследуются правила

редактирования и удаления комментариев, которые можно задать ниже таблицы "Добавление комментариев". Данные настройки могут регулировать действия отличные от "добавления" и имеют следующие значения: "запретить", "разрешить" и "разрешить если нет ответов". Рассмотрим эти настройки более подробно.

Значение "запретить" не позволяет пользователям редактировать или удалять (зависит от того, где значение установлено) свои комментарии.

Значение "разрешить" позволяет пользователям редактировать или удалять (зависит от того, где значение установлено) свои комментарии.

Значение "разрешить, если нет ответов" позволяет пользователям редактировать или удалять (зависит от того, где значение установлено) свои комментарии, но только при условии если комментарий остался неотвеченным.

Аналогично каталогу, комментирование включается или выключается для разделов и компонентов в разделе (вкладка "Настройки").

Важное замечание: вне зависимости от указанных выше параметров, типы прав "Директор", "Супервизор" и "Редактор" (с установленной опцией "модерирование") — могут добавлять, редактировать и удалять комментарии и ответы на них.

В зависимости от вышеописанных настроек, блок вывода комментария или ответа на комментарий, может содержать или не содержать ссылки на ответ, редактирование или удаление комментария.

Для подключения комментариев к компоненту, следует произвести несложные действия.

В системных настройках компонента, в любом месте, следует прописать код:

```
$nc_comments = new nc_comments($cc);
```

Этот код привяжет текущий компонент в разделе к функционалу комментариев и инициализирует для этого отдельный объект. Компонентов в разделе может быть два и больше, к каждому можно подключить функционал комментариев, даже если планируется отображать сразу все эти

компоненты на одной странице. В этом случае будет создано несколько подобных объектов.

Для вывода количества комментариев на определённое сообщение, следует в поле "Объект в списке" или "Отображение объекта" добавить следующий код:

```
<b>комментарии:</b>".$nc_comments->count($f_RowID) ."
```

Этот метод выведет уже заранее подсчитанные значения, что гораздо быстрее пересчёта "на лету".

Вывод "стенки комментариев" и ссылки на добавление осуществляется добавлением следующего кода в поле "Отображение объекта":

```
<div>".$nc_comments->wall($f_RowID) . "</div>
```

Этот же блок можно добавить и в поле "Объект в списке", но добавлять комментарии непосредственно в выводе списка объектов, представляется весьма специфическим функциональным решением.

Кастомизация блоков отображения и более подробное описание методов, будут рассмотрены ниже.

Функционал модуля

Функционал модуля выполнен в объектно-ориентированном стиле и содержит несколько основных методов, являющимися методами класса **nc_comments**.

Создание объекта `$nc_comments` производит следующий код:

```
$nc_comments = new nc_comments($cc) ;
```

Где `$cc` идентификатор компонента в объекте. Данная команда должна выполняться в системных настройках компонента. Имя переменной, в данном случае `$nc_comments`, может быть любым.

Методы класса:

```
$nc_comments->wall($message_id, $template = 0, $from = 0, $quantity = 0, $reset = 0, $ignore = 0, $show_all = 0);
```

Метод выводит «стенку комментариев» в прикрепленном объекте. Под прикрепленным объектом, понимается сообщение, которое комментируют. В методе доступны следующие параметры:

\$message_id – идентификатор сообщения, к которому следует вывести комментарии.

\$template – идентификатор шаблона вывода, прописанного в административной части модуля (по умолчанию равен 0)

\$from – параметр, обозначающий с какого комментария начинать вывод «стенки комментариев» (по умолчанию равен 0).

\$quantity – параметр, указывающий сколько комментариев вывести (по умолчанию равен 0, что означает, что нет ограничения на кол-во комментариев для вывода).

С помощью параметров **\$from** и **\$quantity** организована страничная разбивка комментариев.

\$reset – параметр, с помощью которого можно принудительно произвести инициализацию массива комментариев, если на момент использования метода он пуст. Значение по умолчанию – 0.

\$ignore – параметр, используемый для организации листинга (т.е. страничной разбивки) комментариев. При значении 1 исключает из вывода подключение javascript файла comments.js, повторное формирование json параметров, префикс и суффикс «стенки комментариев», дублирование листинга комментариев и кнопки показа всех комментариев (если она включена).

\$show_all – параметр, используемый для обнуления значения параметра **\$quantity** для работы кнопки «Показать все».

```
$nc_comments->count($message_id, $selector = 0);
```

Метод выводит количество комментариев в прикрепленном объекте. Доступны следующие параметры:

\$message_id – идентификатор сообщения, в котором следует подсчитать комментарии,

\$selector – параметр имеет 3 варианта значений:

0 - считать и комментарии, и ответы (по умолчанию),

1 - считать только комментарии,

2 - считать только ответы.

```
$nc_comments->last_visit($message_id);
```

Метод обновляет в таблице `Comments_LastVisit` сведения о последнем посещении текущего пользователя страницы объекта с комментариями. В методе доступен только 1 параметр:

\$message_id – идентификатор сообщения, сведения о посещении которого следует обновить.

```
$nc_comments->addComment($message_id, $parent_message,  
$comment, $user = 0, $nc_comments_guest_name = '',  
$nc_comments_guest_email = '');
```

Метод добавляет комментарий в базу и пересчитывает общее количество комментариев и ответов в таблице `Comments_Count`. В методе доступны параметры:

\$message_id – идентификатор сообщения, в котором добавлен комментарий,
\$parent_message – идентификатор родителя комментария,
\$comment – текст комментария,
\$user – идентификатор пользователя, добавившего комментарий,
\$nc_comments_guest_name – имя неавторизованного пользователя, добавившего комментарий,
\$nc_comments_guest_email – электронная почта неавторизованного пользователя, добавившего комментарий.

Последние 2 параметра заполняются, если в настройках модуля выставлены галочки, разрешающие неавторизованным пользователям вводить имя и email (см. подробнее в параграфе «Административная часть»).

```
$nc_comments->updateComment($id, $comment, $user = 0);
```

Метод обновляет сведения о комментарии в базе. Доступны параметры:

\$message_id – идентификатор обновляемого комментария,
\$comment – текст комментария,
\$user – идентификатор пользователя, изменившего комментарий.

```
$nc_comments->deleteComment($id);
```

Метод удаляет комментарий из базы по заданному идентификатору (**\$id**).

```
$nc_comments->changeChecked($id, $action='');
```

Метод включает/выключает комментарий. Доступны параметры:

\$id – идентификатор включаемого/выключаемого комментария,
\$action – параметр по умолчанию не имеет явно заданного действия, поэтому применение данного метода включит комментарий, если он выключен и наоборот. Дополнительные значения устанавливают действия вне зависимости от текущего значения поля **Checked** таблицы **Comments_Text**:

Check – включить комментарий,
Uncheck – выключить комментарий.

```
$nc_comments->delete_allComments();
```

Метод удаляет из базы все комментарии и записи таблицы **Comments_Count**.

Все параметры методов, у которых есть значение по умолчанию, являются необязательными для заполнения.

○ Административная часть

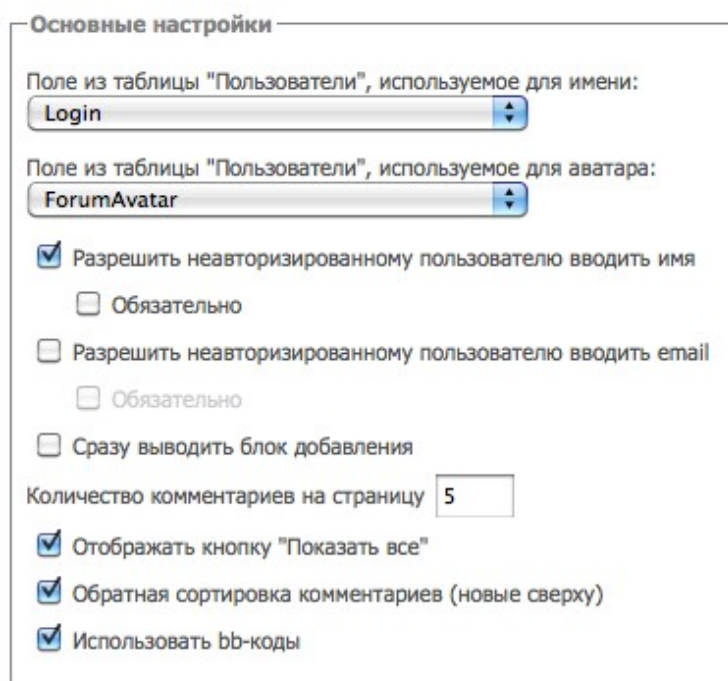
Панель управления модулем («Настройки» – «Список модулей» – «Комментарии») поделена на 2 вкладки: «Настройки» и «Комментарии». Первая вкладка включает в себя вкладки по управлению общими настройками, шаблоном вывода, а также подписками и оповещениями, вторая – по управлению всем списком комментариев, конвертером из старых версий и средствами по оптимизации данных.

Настройки

Основные настройки

Вкладка «Общие настройки» состоит из трех блоков. В первом («Основные настройки») настраиваются:

- поля из таблицы «Пользователи», используемые для имени и аватара зарегистрированного пользователя;
- возможность ввода незарегистрированным пользователем имени и email, причем при установке галочек «Обязательно» у каждого из эти пунктов будет выводиться ошибка над формой добавления комментария с просьбой заполнить поля для ввода имени и email;
- возможность сразу, т.е. без ссылки «оставить комментарий», показывать форму добавления комментария;
- количество комментариев на страницу. При цифре, отличной от нуля, автоматически включается листинг комментариев, причем количество комментариев на страницу не учитывает ответы. Т.е. если у нас всего 11 комментариев, из них 3 — это ответы к комментарию номер 1, а значение в поле «количество комментариев на страницу» равно 5, то на первой странице листинга будет фактически 8 комментариев (5 + 3 ответа на первый), и 3 комментария на второй.
- возможность показа кнопки «Показать все» (активна при включённом листинге комментариев),
- возможность включения обратной сортировки комментариев (т.е. последние комментарии располагаются вверху списка),
- возможность использования bb-кодов в комментариях.



Основные настройки

Поле из таблицы "Пользователи", используемое для имени:

Поле из таблицы "Пользователи", используемое для аватара:

☒ Разрешить неавторизированному пользователю вводить имя
☐ Обязательно

☐ Разрешить неавторизированному пользователю вводить email
☐ Обязательно

☐ Сразу выводить блок добавления

Количество комментариев на страницу

☒ Отображать кнопку "Показать все"

☒ Обратная сортировка комментариев (новые сверху)

☒ Использовать bb-коды

Рис. 2

Во втором блоке («Премодерация») предложено выбрать степень премодерации для добавляемых комментариев. Положение «нет» означает, что премодерация отключена; «только для неавторизованных

пользователей» — соответственно, добавляет комментарий выключенным для неавторизованного пользователя/включённым для авторизованного; «всегда» — означает премодерацию для всех пользователей, кроме пользователей с правами директора, супервизора и редактора с возможностью модерирования.

На месте комментария, попадающего под премодерацию, будет выведено сообщение, которое настраивается во вкладке «Шаблон вывода» в поле «Сообщение о премодерации комментариев».

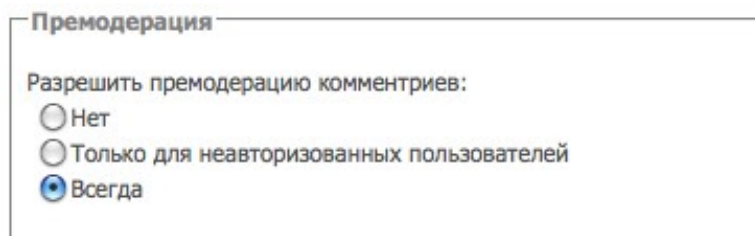


Рис. 3

Последний блок «Новые комментарии» состоит из двух настроек:

- подсветка новых комментариев (цвет подсветки настраивается в шаблоне вывода с учётом макропеременной %IS_NEW),
- возможность показа кнопки перехода к следующему новому комментарию (не активна при включённом листинге комментариев).

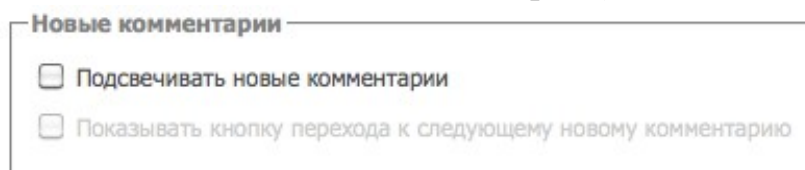


Рис. 4

Шаблон вывода

В этой вкладке можно добавить различные шаблоны оформления блоков комментирования. Решение о том, какой шаблон использовать, система принимает на основе идентификатора, переданного в функции вывода «стенки комментариев» или же, если такого идентификатора задано не было, использует шаблон по умолчанию. Это позволяет кастомизировать внешний вид вывода комментариев под различные задачи. Рассмотрим настройки подробнее.

В первом блоке «Шаблоны вывода комментариев» вынесен выпадающий список предустановленных шаблонов системы. Для добавления нового шаблона следует выбрать последний пункт списка "новый шаблон вывода".

Если были внесены изменения в настройки текущего выбранного шаблона — следует нажать кнопку "Сохранить", иначе настройки не будут сохранены.

В блоке «Настройки шаблона вывода» устанавливается внешний вид всех блоков комментирования. Почти в каждом блоке присутствуют свои макропеременные, которые система обрабатывает при выводе блока. Например, макропеременная `%USER_ID` в блоке "Блок вывода комментария" превратится при выводе в идентификатор автора комментария. Для блоков, в которых есть подобные макропеременные, будет указана таблица с именами этих переменных. Рассмотрим все настройки шаблона вывода поближе:

- использовать по умолчанию — позволяет назначить выбранный шаблон шаблоном, используемым по умолчанию;
- название шаблона — любое «говорящее» название шаблона, например «шаблон для новостей»;
- префикс «стенки комментариев» — позволяет разместить любой HTML-код в самом начале блока вывода «стенки комментариев». Это может быть какой либо заголовок или описание правил. В этом блоке доступна только 1 макропеременная, и она обязательная:

%ID – идентификатор «стенки комментариев»

Замечание: здесь и далее по тексту жирным шрифтом выделены обязательные переменные

- блок вывода комментария — этот блок определяет, как следует выводить текст комментария. Ниже список макропеременных, доступных в этом блоке:

%ID – идентификатор комментария

%COMMENT – текст комментария

%IS_NEW – если комментарий новый, псевдопеременная равна 1

%USER_ID – идентификатор автора комментария из таблицы User

%USER_NAME – поле Login автора из таблицы User

%USER_AVATAR – ссылка на аватар пользователя

%GUEST_NAME – имя гостя, если при добавлении комментария он его ввёл

%GUEST_EMAIL – email гостя, если при добавлении комментария он его ввёл

%DATE – дата добавления комментария

%UPDATED – дата обновления комментария (редактирование)

%REPLY_LINK – ссылка ответа на комментарий, заполняется из блока «Ссылка на ответ»

%EDIT_LINK – ссылка на редактирование комментария, заполняется из блока «Ссылка на редактирование»
%DELETE_LINK – ссылка на удаление комментария, заполняется из блока «Ссылка на редактирование»
%REPLY_BLOCK – указывает, в каком месте выводить блок ответов на комментарий
%LEVEL – уровень вложенности ответа, для комментариев равен 0

●блок вывода ответа на комментарий — этот блок определяет, как следует выводить текст ответа на комментарий. В этом блоке можно задать отступ от левого края, чтобы показать, что данный комментарий является ответом. Макропеременные аналогичны тем, что указаны выше для «Блока вывода комментария».

●ссылка на комментирование — данная настройка определяет, что следует выводить в ссылке на комментирование. *Изменение положения ссылки в «стенке комментариев» не предусмотрено.* Ниже список макропеременных, доступных в этом блоке:

%ID – идентификатор ссылки
%ACTION – действие, например onclick="%ACTION"
%TEXT – текст ссылки или картинка

●ссылка на ответ — данная настройка определяет, что следует выводить в ссылке ответа на комментарий. Макропеременные аналогичны блоку «Ссылка на комментирование», описанному выше.

●ссылка на редактирование — данная настройка определяет, что следует выводить в ссылке на редактирование комментария. Макропеременные аналогичны блоку «Ссылка на комментирование», описанному выше.

●ссылка на удаление — данная настройка определяет, что следует выводить в ссылке на удаление комментария. Макропеременные аналогичны блоку «Ссылка на комментирование», описанному выше.

●блок добавления комментария или ответа — этот блок отображает форму добавления комментария. Ниже таблица макропеременных, доступных в этом блоке:

%FORM_ID – идентификатор формы добавления
%TEXTAREA_ID – идентификатор текстового поля
%TEXTAREA_VALUE – значение внутри текстового поля
%CANCEL_BUTTON_ID – идентификатор кнопки «Отменить»
%CANCEL_BUTTON_ACTION – действие кнопки «Отменить»
%SUBMIT_BUTTON_ID – идентификатор кнопки «Сохранить»

`%ISBBCODES` – логическая переменная, true, если BB-коды разрешены

Поля для ввода гостем имени и email выводятся следующим кодом:

```
<span id='nc_comments_guest_name'> Ваше имя  
<input type='text' name='nc_comments_guest_name'></span>  
<span id='nc_comments_guest_email'> Ваш email  
<input type='text' name='nc_comments_guest_email'></span>
```

Важно: атрибуты id и name у тегов span и input должны точно соответствовать приведённым выше.

- блок редактирования комментария или ответа — этот блок отображает форму редактирования комментария. Макропеременные аналогичны блоку «Блок добавления комментария или ответа», описанному выше.

- блок удаления комментария или ответа — этот блок отображает форму удаления комментария. Все макропеременные этого блока обязательны:

- `%FORM_ID` – идентификатор формы добавления
 - `%CANCEL_BUTTON_ID` – идентификатор кнопки «Отменить»
 - `%CANCEL_BUTTON_ACTION` – действие кнопки «Отменить»
 - `%SUBMIT_BUTTON_ID` – идентификатор кнопки «Сохранить»

Следует отметить, что блоки «Блок вывода комментария», «Блок вывода ответа на комментарий», «Блок добавления комментария или ответа», «Блок редактирования комментария или ответа» и «Блок удаления комментария или ответа» обрабатываются PHP функцией eval(), что позволяет использовать PHP код внутри этих блоков.

- суффикс «стенки комментариев» — позволяет разместить любой HTML-код в самом конце блока вывода «стенки комментариев».

- шаблон вывода листинга комментариев — позволяет кастомизировать внешний вид ссылок «следующая страница» и «последняя страница». Дополнительно в макете дизайна можно определить массив `$browse_comments` такого вида:

```
$browse_comments['prefix'] = "";  
$browse_comments['suffix'] = "";  
$browse_comments['active'] = "<b>%PAGE</b>";  
$browse_comments['unactive']="<a href='#'  
onclick='%NAV_ACTION'>%PAGE</a>";  
$browse_comments['divider'] = " | ";
```

●кнопка показа всех комментариев — задаёт внешний вид кнопки/ссылки показа всех комментариев. Доступна только 1 макропеременная:

%SHOW_ALL — действие кнопки «Показать все»

●сообщение об ошибке — определяет внешний вид сообщения об ошибке. Сообщения об ошибке выводятся, когда заданы настройки, требующие ввод гостем имени и email (т.е. также проставлены галочки «обязательно»), но гость не заполнил одно или оба поля, не правильно введена катча и другие ошибки. Доступна только 1 макропеременная:

%WARNTEXT — сообщение об ошибке

●сообщение о премодерации комментариев — определяет внешний вид сообщения. Сообщение выводится, когда в настройках включена премодерация для комментариев, добавляемых неавторизованными пользователями или для всех. Макропеременные отсутствуют.

●кнопка перехода к новому комментарию — задаётся внешний вид кнопки. *Кнопка автоматически не активна при включённом листинге комментариев.* Доступна 1 макропеременная:

%NEW_COMMENT_BUTTON_ACTION — действие кнопки

Подписки и рассылки

Во вкладке находится управление всем, что связано с подпиской на новые комментарии:

Доступна подписка на комментарии — пользователи смогут подписаться на новые комментарии.

Ссылка на подписку — данная настройка определяет, что следует выводить в ссылке на подписку. Доступные макропеременные в этом поле:

%ID — идентификатор ссылки

%ACTION — действие, например `onclick='%ACTION'`

%TEXT — текст ссылки

Тема письма — шаблон темы письма-уведомления о новом комментарии.

Шаблон тела письма — шаблон письма, приходящий пользователям при новом комментарии. В теме и теле шаблона письма доступны следующие переменные:

%SUBDIVISION_NAME — название раздела с комментариями

%COMMENT_TEXT — текст комментария
 %FULL_LINK — адрес страницы с комментарием
 \$comment_id — номер комментария
 \$comment_text — текст комментария
 \$message_id — номер комментируемого объекта
 \$subdivision_name — имя раздела с комментариями
 \$subdivision_id — номер раздела
 \$sub_class_name — имя компонента в разделе
 \$sub_class_id — номер компонента в разделе
 \$fullLink — адрес страницы с комментарием

Комментарии

Список комментариев

На странице списка комментариев вы можете увидеть все комментарии, организованные в таблице с возможностью осуществления выборки по ним, сортировки и разбивки на страницы.

Комментарии

Комментарии | Настройки

Список комментариев | Конвертер из старых версий | Оптимизация данных

▼ Выборка комментариев:

По разделу:

По компоненту:

☒ все ☐ включенные ☐ выключенные

Количество комментариев на страницу:

ID	Автор	Текст комментария	Действия	☒
13	admin	Согласен!	выключить изменить	☐
12	admin	Сложно судить о том, чего не понимаешь	выключить изменить	☐
11	admin	Не думаю, что это так	выключить изменить	☐
10	admin	Присутствие на вершинах многих подводных гор наложенных друг на друга построек вовсе не означает, что отложение затруднено.	выключить изменить	☐
9	Астроном	Согласен, что оттаивание пород разогревает конус выноса, но не считаю, что лавовый купол причленяет к себе оливин.	выключить изменить	☐
8	bob	А можно и не ожидать	выключить изменить	☐
7	admin	Можно ожидать, что перенос причленяет к себе горст, делая этот типологический таксон районирования носителем важнейших инженерно-геологических характеристик природных условий.	выключить изменить	☐

Рис. 5

Текст каждого комментария можно изменить (при нажатии на ссылку «изменить»), можно просмотреть данные добавившего комментарий авторизованного пользователя (данные о гостях выводятся в зависимости от введенной ими (гостями) информации), а также по ссылке текста комментария можно перейти к нему непосредственно на самом сайте.

Удалить и включить/выключить комментарий можно по одному или сразу несколько, отметив галочками в последней колонке.

Если комментариев в списке много и искать нужный долго, можно воспользоваться выборкой (поиском). Для этого нажмите на ссылку «Выборка комментариев».

В появившейся форме заполните параметры выборки и нажмите кнопку «Осуществить выборку». Вам будут показаны все комментарии, подходящие под поисковый запрос.

Конвертер из старых версий

Конвертер предназначен для переноса информации из стандартных «компонентных» комментариев в базу данных модуля. При этом учитываются только стандартные поля компонента:

User_ID — системное поле компонента, содержащее идентификатор автора комментария

Created — системное поле компонента, содержащее дату создания комментария

Message — текст комментария

Object_Sub_Class_ID — идентификатор компонента в разделе, в котором находится комментируемый объект

Object_Parent_ID — идентификатор родительского комментария

Object_ID — идентификатор комментируемого объекта

После конвертирования в базе данных модуля будет содержаться информация из компонентных комментариев, которую можно выводить или править, используя функционал модуля.

Перед конвертированием рекомендуется отключить функционал добавления комментариев через "компонентный" функционал, это позволит избежать потери информации, добавленной в процессе конвертации.

На первом этапе конвертации предлагается выбрать сайт, содержащий нужный раздел с комментариями (имеется в виду, что к разделу прикреплен компонент комментариев).

После выбора сайта появится диалог выбора упомянутого выше раздела.

Заключительный этап — выбор компонента в разделе, этот шаг необходим на случай нескольких прикрепленных компонентов в разделе.

После успешного окончания процесса конвертации, можно убрать строки, отвечающие за "компонентный" функционал комментариев из компонентов, к которым он был подключен. Взамен следует подключить функционал модуля, описанного в данной документации.

Оптимизация данных

На этой вкладке вы можете пересчитать количество комментариев и ответов на них в таблице `Comments_Count`, если по каким-то причинам вам кажется, что в данный момент это количество подсчитано неправильно.

После установления галочки *«Пересчитать комментарии и ответы»* и нажатия кнопки *«Оптимизировать»* вам будет выведено в скольки объектах пересчитаны комментарии.

• **Модуль «Логирование»**

○ **Описание**

Модуль предназначен для отображения всех изменений в системе (добавление, изменение, удаление), которые совершаются пользователями или скриптами через стандартный функционал.

Информация получается из системных событий и сохраняется в базе данных. Просмотреть её можно в административной части модуля.

Стоит отдельно подчеркнуть, что логируются только изменения, выполненные через API системы. Если кто либо добавил, зменил или удалил данные посредством прямых MySQL запросов, самописных скриптов и т.п., то информация об этом сохранена не будет.

○ **Административная часть**

▪ **Последние изменения**

В панели управления модуля ("Настройки" — "Список модулей" — "Логирование") можно просмотреть всю доступную информацию о прошедших событиях.

В каждом событии присутствуют внутренние ссылки на объекты, подвергшиеся изменению. Следует понимать, что если объект будет или был удалён, либо перемещён — ссылка будет бесполезной.

Список событий можно сортировать по различным критериям. Для этого достаточно выбрать параметры сортировки и нажать кнопку «Отобразить» в нижнем правом углу экрана. На данный момент возможно сортировать события по дате, названия, пользователю, при этом выбирая порядок сортировки. Кроме того, возможно использовать фильтр по типам событий (добавление, изменение, удаление), по сущностям системы (сайт, раздел, компонент и т.п.), чтобы отображать только интересующие записи.

Статистика логирования разбивается на страницы и выделяется цветом для удобства восприятия.

Для очистки данных логирования достаточно нажать кнопку «Очистить». После этого следует подтвердить намерение повторным нажатием на кнопку «Очистить» в правом нижнем углу экрана, либо нажать на кнопку «Отмена».

■ Константы модуля

Важно: данные настройки предназначены только для опытных пользователей, установка значений констант отличных от значений по-умолчанию может привести к некорректной работе модуля в целом.

В административном списке предопределённых констант модуля ("Настройки" — "Настройка модуля" — "Логирование" — "Настройки") по-умолчанию прописаны следующие обязательные константы:

●**ACTIVITY** — Включить/Выключить логирование. Включенное логирование будет лишним, если над проектом работает только один пользователь.

●**ROWS_PER_PAGE** — Количество отображаемых на странице элементов.

●**PAGES_NUMBER** — Количество отображаемых страниц.

Модуль «Форум»



Принцип работы модуля

Функционал форума основан на базовом функционале системы. Это подразумевает, что ввод и вывод данных осуществляется посредством компонентов форума, в которых и находится основная часть кода. Макеты дизайна системы не требуют никаких доработок для обеспечения нормального функционирования форума. Дополнительные расширения форума в макетах (вывод списка последних сообщений, количества форумов и т.п.) добавляются как обычные компонентные расширения.

Такой подход позволяет максимально гибко настраивать форум без изменения ядра системы и файлов модуля. Все основные запросы к БД (базе данных) находятся в системных настройках компонентов, что даёт возможность расширять их и оптимизировать под конкретные задачи проекта. Нет никаких ограничений на изменение внешнего вида форума.

Для повышения производительности в БД находится несколько таблиц, которые хранят информацию о категориях и разделах форума, количестве топиков и ответов, просмотров на них и т.п. Данные попадают в эти таблицы при трансляции системных событий, которые затрагивают функционал форума. Например, при добавлении сообщения (стандартный функционал системы) в раздел форума обновляется информация в БД о количестве сообщений на форуме.

Для упрощения работы с данными из БД в модуле предусмотрен определённый набор функционала, о котором будет рассказано ниже.

○ Подключение функционала форума

Для того чтобы на сайте появился новый форум, следует выполнить несколько несложных действий. Основное действие — это добавление корневого раздела форумов (раздел в котором находятся все форумы, и таких разделов может быть несколько). Корневой раздел форумов может так же называться родительским или корневым форумом, все эти определения синонимичны.

Чтобы создать новый корневой раздел форумов, достаточно в административной панели системы добавить новый раздел и подключить к нему компонент «Список форумов». *Стоит заметить, что в этом случае параметр TRACKING_MODE в настройках модуля не должен быть равен 0, подробности в разделе «Константы модуля».* После этого, система автоматически идентифицирует этот раздел, как относящийся к модулю «Форум». Если требуется создать трансляцию списка форумов в данном разделе посредством RSS, это сделать весьма несложно. Достаточно в созданный раздел добавить ещё один компонент «Список форумов», указав при этом, что для отображения следует использовать шаблон компонента «RSS».

Следует отметить, что функционал форума не поддерживает модуль "Кэширование". Отображение многих данных зависит от привилегий пользователя (ссылки на редактирование сообщений, назначение прав пользователям и т.п.). Ко всему прочему, издержки на поддержание актуальности кэша столь динамичного контента слишком велики.

Далее можно приступить непосредственно к добавлению форумов в раздел. Сделать это можно как минимум двумя разными способами:

1. Зайти на сайт (front-office), найти раздел форум и нажать на ссылку "Добавить форум" (при условии достаточных прав — *администратор корневого раздела*, отобразится форма добавления). В появившейся форме нужно ввести информацию о новом форуме.

2. В административной панели системы следует добавить новый подраздел в корневой раздел форумов. Далее следует подключить к разделу сначала компонент «Топики», а затем компонент «Ответы». Присвоив каждому легкоидентифицируемые ключевые слова, например, topic и reply, соответственно. *Стоит заметить, что в этом случае параметр TRACKING_MODE в настройках модуля не должен быть равен 0, подробности в разделе «Константы модуля».* Система автоматически идентифицирует этот раздел как относящийся к модулю «Форум». По аналогии с корневым разделом для трансляции RSS в раздел форума добавляются шаблоны компонентов «Топики RSS» и «Ответы RSS».

Описанных действий достаточно для начала работы форума.

○ Разграничение прав

Права на разделы форума назначаются и учитываются так же, как и на любые другие разделы системы. Если установить право на чтение раздела конкретного форума под названием «Зарегистрированные», то содержимое раздела смогут увидеть только зарегистрированные пользователи. Описаное актуально и для компонентов в разделе.

В каждом форуме можно также указать, кто может добавлять топики «Важное», «Объявление», а так же закрывать и открывать топики. Для этого следует выбрать раздел форума в карте сайта, перейти к списку используемых компонентов, выбрать настройки компонента в разделе «Топики» (название может отличаться, если при создании было указано другое) и перейти к «Визуальным настройкам компонента». Далее из списка следует выбрать желаемое значение.

Может возникнуть ситуация, когда наряду с обычными форумами в разделе подразумевается наличие форума, который могут видеть только определённые пользователи. Список форумов формируется запросом к БД MySQL в компоненте «Список форумов», можно модифицировать этот запрос, но это замедлит формирование списка. Альтернатива данному способу — создать отдельный раздел с форумами, которые будут видны определённым пользователям или группам пользователей.

○ Функции модуля

Функционал модуля выполнен в объектно-ориентированном стиле. В форуме присутствуют 4 основные сущности, которые связаны с компонентами форума:

- 1.Объект списка форумов. По сути, это объект управления корневыми разделами форумов. *Связанный компонент: "Список форумов".*
- 2.Объект форумов. Содержит в себе методы для получения информации о форумах, а также методы модификации этой информации. *Связанный компонент: "Топики". Данный компонент даёт понять системе, что раздел относится к модулю "Форум", поэтому он и связан с этим объектом и объектом топиков.*

3.Объект топиков. Содержит в себе методы для получения информации о топиках, а также методы модификации данной информации. *Связанный компонент: "Топики"*.

4.Объект ответов. Содержит в себе методы для получения информации об ответах на топики, а также методы модификации данной информации. *Связанный компонент: "Ответы"*.

Ниже приведён список доступных методов для каждого объекта. Все описанные методы относятся к публичному уровню доступа (public).

■

■ **Объект списка форумов**

●**get_object()**. Статичный метод. Для получения объекта следует использовать следующую конструкцию:

```
$parent_obj = nc_forum2_parent::get_object();
```

Метод возвращает объект списка форумов.

Если функционал предполагается использовать в компонентах, оптимальное место для написания этого кода — системные настройки компонента.

●**get_class_id()**. Метод получения идентификатора связанного компонента:

```
$parent_obj->get_class_id();
```

Метод возвращает числовой идентификатор.

●**get_rss_class_id()**. Метод получения идентификатора связанного шаблона компонента для RSS:

```
$parent_obj->get_rss_class_id();
```

Метод возвращает числовой идентификатор.

●**get_subclass_id()**. Метод получения идентификатора компонента в разделе «Список форумов»:

```
$parent_obj->get_subclass_id($sub, $template = 0);
```

Первый обязательный параметр метода определяет, из какого корневого форума (раздела) следует получить данные. Второй параметр — идентификатор шаблона компонента.

Метод возвращает числовой идентификатор.

●**get_rss_subclass_id()**. Метод получения идентификатора компонента в разделе «Список форумов», который используется для RSS (в данном случае это шаблон «RSS» компонента «Список форумов»):

```
$parent_obj->get_subclass_id($sub);
```

Единственный обязательный параметр метода определяет, из какого корневого форума (раздела) следует получить данные.

Метод возвращает числовой идентификатор.

По сути, метод использует функционал метода get_subclass_id() с передачей второго параметра.

●**get_rss_url()**. Метод получения ссылки на ленту RSS для списка форумов:

```
$parent_obj->get_rss_url($sub);
```

Единственный обязательный параметр метода определяет, из какого корневого форума (раздела) нужно получить данные. В данном случае это идентификатор раздела в карте сайта.

Метод возвращает строку.

Ссылка будет получена только в том случае, если к корневому разделу прикреплен шаблон «RSS» компонента «Список форумов».

●**get_data()**. Метод получения данных корневого раздела из MySQL таблицы Forum_Subdivisions:

```
$parent_obj->get_data($id, $item = "");
```

Первый обязательный параметр метода определяет, из какого корневого форума (раздела) необходимо получить данные. В данном случае это идентификатор раздела в карте сайта. Второй параметр — ключевое слово конкретного поля из записи.

Если второй параметр не указан, метод вернёт ассоциативный массив данных или конкретное значение поля.

•**count_user_messages()**. Метод подсчёта количества сообщений, написанных пользователем во всех форумах корневого раздела или всех написанных в системе сообщений:

```
$parent_obj->count_user_messages($user_id, $sub=0, $type=0);
```

Первый обязательный параметр метода определяет для какого пользователя подсчитываются сообщения. Второй параметр определяет, из какого корневого форума (раздела) необходимо получить данные.

Третий параметр определяет тип подсчитываемых сообщений (0 — все сообщения, 1 — только топики, 2 — только ответы).

Метод возвращает числовое значение.

•**get_users()**. Метод получения идентификаторов всех пользователей, которые оставляли посты во всех форумах корневого раздела или во всех форумах системы:

```
$parent_obj->get_users($sub = 0, $type = 0);
```

Первый необязательный параметр определяет из какого корневого форума (раздела) получить данные. Если первый параметр не указан — учитываются все корневые разделы. Второй параметр определяет тип подсчитываемых сообщений (0 — все сообщения, 1 — только топики, 2 — только ответы).

Метод возвращает массив идентификаторов пользователей.

•**set_description()**. Метод обновления описания корневого раздела в MySQL таблице Forum_Subdivisions:

```
$parent_obj->set_description($parent,  
$description);
```

Первый обязательный параметр — идентификатор корневого форума (раздела). В данном случае это идентификатор раздела в карте сайта. Второй параметр — текстовое описание корневого форума.

▪ Объект форумов

• **get_object()**. Статичный метод. Для получения объекта следует использовать следующую конструкцию:

```
$forum_obj = nc_forum2_forum::get_object();
```

Метод возвращает объект форумов.

Если функционал предполагается использовать в компонентах, оптимальное место для написания этого кода — системные настройки компонента.

• **get_class_id()**. Метод используется для получения идентификатора связанного компонента:

```
$forum_obj->get_class_id();
```

• **get_rss_class_id()**. Метод получения идентификатора связанного шаблона компонента для RSS:

```
$parent_obj->get_rss_class_id();
```

Метод возвращает числовой идентификатор.

• **get_subclass_id()**. Метод получения идентификатора компонента в разделе «Топики» (для форумов используется именно этот компонент):

```
$parent_obj->get_subclass_id($sub, $template = 0);
```

Первый обязательный параметр метода определяет, из какого форума (раздела) следует получить данные. Второй параметр —

идентификатор шаблона компонента, если нужно получить идентификатор компонента в разделе с шаблоном компонента.

Метод возвращает числовой идентификатор.

•**get_rss_subclass_id()**. Метод получения идентификатора компонента в разделе «Топики», который используется для RSS (в данном случае это шаблон «RSS» компонента «Топики»):

```
$parent_obj->get_subclass_id($sub);
```

Единственный обязательный параметр метода определяет, из какого форума (раздела) следует получить данные.

Метод возвращает числовой идентификатор.

По сути, метод использует функционал `get_subclass_id()`, с передачей второго параметра.

•**get_views()**. Метод используется для получения количества просмотров форума:

```
$forum_obj->get_views($forum_id);
```

Единственный обязательный параметр метода определяет, из какого форума следует получить данные, в качестве идентификатора форума подразумевается идентификатор раздела конкретного форума в карте сайта.

Метод возвращает числовое значение.

•**get_forum_url_by_cc_id()**. Метод используется для получения ссылки на раздел форума по идентификатору компонента в разделе:

```
$forum_obj->get_forum_url_by_cc_id($cc);
```

Единственный обязательный параметр функции определяет, по какому компоненту в разделе следует получить результат. В данную функцию стоит передавать идентификатор компонента в разделе "Топики". *Результирующая ссылка формируется с учётом отображения во front-office и back-office.*

Метод возвращает строку.

Данный метод будет полезен для формирования возвратной ссылки в альтернативных формах.

•**get_rss_url()**. Метод получения ссылки на ленту RSS списка топиков:

```
$parent_obj->get_rss_url($sub);
```

Единственный обязательный параметр метода определяет, из какого форума (раздела) получить данные. В данном случае это идентификатор раздела в карте сайта.

Метод возвращает строку.

Ссылка будет получена только в том случае, если к разделу форума прикреплен шаблон «RSS» компонента «Топики».

•**count_user_messages()**. Метод используется для получения количества сообщений пользователя:

```
$forum_obj->count_user_messages($user_id, $forum=0, $type=0);
```

В функцию передаётся 3 параметра:

- 1.\$user_id - идентификатор пользователя (*обязательный*);
- 2.\$forum - идентификатор раздела форума;
- 3.\$type - тип получаемого ответа:
(0 — все сообщения, 1 — только топики, 2 — только ответы);

Метод возвращает числовое значение.

•**get_users()**. Метод получения идентификаторов всех пользователей, оставивших посты в форуме:

```
$parent_obj->get_users($sub = 0, $type = 0);
```

Первый необязательный параметр определяет, из какого форума (раздела) следует получить данные. Второй параметр определяет тип

сообщений (0 — все сообщения, 1 — только топики, 2 — только ответы).

Метод возвращает массив идентификаторов пользователей.

•**get_data()**. Метод получения данных раздела форума из MySQL таблицы Forum_Subdivisions:

```
$parent_obj->get_data($id, $item = "");
```

Первый обязательный параметр метода определяет, из какого форума (раздела) получить данные. В данном случае это идентификатор раздела в карте сайта. Вторым параметром — ключевое слово конкретного поля из записи.

Если вторым параметром не указан, метод вернёт ассоциативный массив данных или конкретное значение поля.

•**set_description()**. Метод обновления описания форума в MySQL таблице Forum_Subdivisions:

```
$parent_obj->set_description($parent,  
$description);
```

Первый обязательный параметр — идентификатор форума (раздела). В данном случае это идентификатор раздела в карте сайта. Вторым параметром — текстовое описание форума.

•**set_group()**. Метод привязки форума к определённой категории в MySQL таблице Forum_Groups:

```
$parent_obj->set_group($forum, $group);
```

Первый обязательный параметр — идентификатор форума (раздела). В данном случае это идентификатор раздела в карте сайта. Вторым параметром — идентификатор группы.

Доступные группы можно просмотреть в интерфейсе управления модулем.

●**check_existance()**. Метод проверки существования форума в определённом корневом разделе по ключевому слову:

```
$parent_obj->check_existance($parent_id,  
$keyword) ;
```

Первый обязательный параметр — идентификатор корневого форума (раздела). В данном случае это идентификатор раздела в карте сайта. Второй параметр — ключевое слово раздела (поле EnglishName из таблицы Subdivision).

Метод возвращает идентификатор существующего раздела или *false*.

●**create()**. Метод используется для создания форума (раздела форума с подключёнными компонентами):

```
$forum_obj->create(...);
```

В функцию передаются следующие параметры, их достаточно много, как и работа, которую выполняет данный метод:

- 1.\$parent_id — идентификатор корневого раздела (*обязательный*);
- 2.\$keyword — ключевое слово раздела форума (*обязательный*);
- 3.\$name — название форума (*обязательный*);
- 4.\$checked — включить раздел или нет;
- 5.\$topic_keyword — ключевое слово компонента в разделе «Топики», по умолчанию *topic*;
- 6.\$topic_name — название компонента в разделе «Топики», по умолчанию «Топики»;
- 7.\$reply_keyword — ключевое слово компонента в разделе «Ответы», по умолчанию *reply*;
- 8.\$reply_name — название компонента в разделе «Ответы», по умолчанию «Ответы».
- 9.\$topic_rss_keyword — ключевое слово компонента в разделе «Топики RSS», по умолчанию *topic*;
10. \$topic_rss_name — название компонента в разделе «Топики RSS», по умолчанию «RSS топики»;
11. \$reply_rss_keyword — ключевое слово компонента в разделе «Ответы RSS», по умолчанию *reply*;

12. `$reply_rss_name` — название компонента в разделе «Ответы RSS», по умолчанию «RSS ответы».

Метод возвращает идентификатор созданного форума (раздела).

Объект топиков

•**get_object()**. Статичный метод. Для получения объекта следует использовать следующую конструкцию:

```
$topic_obj = nc_forum2_topic::get_object();
```

Метод возвращает объект топиков.

Если функционал предполагается использовать в компонентах, оптимальное место для написания этого кода — системные настройки компонента.

•**get_class_id()**. Функция используется для получения идентификатора связанного компонента:

```
$topic_obj->get_class_id();
```

Метод возвращает идентификатор компонента.

•**get_rss_class_id()**. Метод получения идентификатора связанного шаблона компонента для RSS:

```
$parent_obj->get_rss_class_id();
```

Метод возвращает числовой идентификатор.

•**get_subclass_id()**. Метод получения идентификатора компонента в разделе «Топики»:

```
$parent_obj->get_subclass_id($sub, $template = 0);
```

Первый обязательный параметр метода определяет, из какого форума (раздела) следует получить данные. Второй параметр — идентификатор шаблона компонента.

Метод возвращает числовой идентификатор.

•**get_rss_subclass_id()**. Метод получения идентификатора компонента в разделе «Топики», который используется для RSS (в данном случае это шаблон «RSS» компонента «Топики»):

```
$parent_obj->get_subclass_id($sub);
```

Единственный обязательный параметр метода определяет, из какого форума (раздела) следует получить данные.

Метод возвращает числовой идентификатор.

По сути, метод использует функционал метода `get_subclass_id()` с передачей второго параметра.

•**get_url()**. Функция используется для получения ссылки на топик по его идентификатору:

```
$topic_obj->get_url($topic_id);
```

Единственный обязательный параметр функции определяет, для какого топика следует получить данные, в качестве идентификатора топика подразумевается идентификатор сообщения. Ссылка формируется с учётом отображения во *front-office* и *back-office*.

Метод возвращает строку.

•**get_rss_url()**. Метод получения ссылки на ленту RSS топика:

```
$parent_obj->get_rss_url($forum_id, $topic_id);
```

Первый обязательный параметр метода определяет, из какого форума (раздела) следует получить данные. В данном случае это идентификатор раздела в карте сайта. Второй параметр — идентификатор топика (сообщения).

Метод возвращает строку.

Ссылка будет получена только в том случае, если к разделу форума прикреплен шаблон «RSS» компонента «Топики».

•**get_views()**. Метод используется для получения количества просмотров топика:

```
$topic_obj->get_views($topic_id);
```

Единственный обязательный параметр метода определяет, из какого топика следует получить данные, в качестве идентификатора топика подразумевается идентификатор сообщения.

Метод возвращает числовое значение.

•**get_closed_status()**. Метод используется для получения статуса топика (открыт/закрит):

```
$topic_obj->get_closed_status($topic_id);
```

Единственный обязательный параметр метода определяет, из какого топика следует получить данные, в качестве идентификатора топика подразумевается идентификатор сообщения.

Метод возвращает булево значение.

•**get_last_ids()**. Метод используется для получения идентификаторов последних топиков:

```
$topic_obj->get_last_ids($sub, $limit = 0);
```

Первый обязательный параметр метода определяет, из какого форума следует получить данные. В данном случае это идентификатор раздела в карте сайта. Второй параметр — количество последних сообщений.

Метод возвращает массив идентификаторов.

▪ Объект ответов

•**get_object()**. Статичный метод. Для получения объекта следует использовать следующую конструкцию:

```
$reply_obj = nc_forum2_reply::get_object();
```

Метод возвращает объект ответов.

Если функционал предполагается использовать в компонентах, оптимальное место для написания этого кода — системные настройки компонента.

•**get_class_id()**. Метод используется для получения идентификатора связанного компонента:

```
$reply_obj->get_class_id();
```

Метод возвращает идентификатор компонента.

•**get_rss_class_id()**. Метод получения идентификатора связанного шаблона компонента для RSS:

```
$parent_obj->get_rss_class_id();
```

Метод возвращает числовой идентификатор.

•**get_subclass_id()**. Метод получения идентификатора компонента в разделе «Ответы»:

```
$parent_obj->get_subclass_id($sub, $template = 0);
```

Первый обязательный параметр метода определяет, из какого форума (раздела) следует получить данные. Второй параметр — идентификатор шаблона компонента.

Метод возвращает числовой идентификатор.

•**get_rss_subclass_id()**. Метод получения идентификатора компонента в разделе «Ответы», который используется для RSS (в данном случае это шаблон «RSS» компонента «Ответы»):

```
$parent_obj->get_subclass_id($sub);
```

Единственный обязательный параметр метода определяет, из какого форума (раздела) следует получить данные.

Метод возвращает числовой идентификатор.

По сути, метод использует функционал метода `get_subclass_id()`, с передачей второго параметра.

•**get_topic_url()**. Метод используется для получения ссылки на топик по идентификатору ответа:

```
$reply_obj->get_topic_url($reply_id);
```

Единственный обязательный параметр метода определяет, для какого ответа получить данные о его топике, в качестве идентификатора ответа подразумевается идентификатор сообщения. *Ссылка формируется с учётом отображения во front-office и back-office.*

Метод возвращает строку.

•**get_topic_id()**. Метод используется для получения идентификатора топика по идентификатору ответа:

```
$reply_obj->get_topic_id($reply_id);
```

Единственный обязательный параметр метода определяет, для какого ответа получить данные о его топике, в качестве идентификатора ответа подразумевается идентификатор сообщения.

Метод возвращает строку.

•**get_curpos_by_topic_id()**. Метод используется для получения страницы, на которой находится последний ответ на топик, по идентификатору топика:

```
$reply_obj->get_curpos_by_topic_id($topic_id);
```

Единственный обязательный параметр метода определяет, для какого топика следует получить данные о странице, на которой находится последний ответ на топик, в качестве идентификатора топика подразумевается идентификатор сообщения.

•**get_last_ids()**. Метод используется для получения идентификаторов последних ответов:

```
$topic_obj->get_last_ids($sub, $limit = 0);
```

Первый обязательный параметр метода определяет, из какого форума следует получить данные. В данном случае это идентификатор раздела в карте сайта. Второй параметр — количество последних сообщений.

Метод возвращает массив идентификаторов.

Информация для разработчиков



▪ Структура таблиц

Для уменьшения нагрузки на БД и упрощения доступа к часто запрашиваемым данным, которые опять же требуют определённых ресурсов и итераций для подсчёта, в модуле используются дополнительные MySQL таблицы.

Рассмотрим структуру каждой из них на примере SQL запросов, которые создают таблицы. Запросы можно использовать для восстановления таблиц (*без данных*).

Forum_Count

В этой таблице хранятся данные форумов о количестве топиков, ответов на топики, идентификаторы последних топиков и ответов.

```
CREATE TABLE `Forum_Count` (  
  `ID` int(11) unsigned NOT NULL auto_increment,  
  `Subdivision_ID` int(11) unsigned NOT NULL default '0',  
  `Topics` int(11) unsigned NOT NULL default '0',  
  `Replies` int(11) unsigned NOT NULL default '0',  
  `Last_Topic_ID` int(10) unsigned NOT NULL default '0',  
  `Last_Reply_ID` int(10) unsigned NOT NULL default '0',  
  PRIMARY KEY (`ID`),  
  KEY `Subdivision_ID` (`Subdivision_ID`),  
  KEY `Last_Topic_ID` (`Last_Topic_ID`),  
  KEY `Last_Reply_ID` (`Last_Reply_ID`)  
) ENGINE=MyISAM DEFAULT CHARSET=cp1251;
```

- `Subdivision_ID` — идентификатор раздела форума;
- `Topics` — количество топиков в данном форуме;
- `Replies` — количество ответов на топики в данном форуме;
- `Last_Topic_ID` — идентификатор последнего добавленного топика;
- `Last_Reply_ID` — идентификатор последнего добавленного ответа.

Forum_Groups

В этой таблице хранятся данные о категориях форумов. Данные из этой таблицы можно использовать при выводе всех категорий форумов.

```
CREATE TABLE `Forum_Groups` (
  `ID` int(10) unsigned NOT NULL auto_increment,
  `Subdivision_ID` int(10) unsigned default NULL,
  `Name` varchar(255) NOT NULL,
  `Description` text NOT NULL,
  `Priority` int(10) default NULL,
  PRIMARY KEY (`ID`),
  KEY `Parent_ID` (`Subdivision_ID`)
) ENGINE=MyISAM DEFAULT CHARSET=cp1251;
```

- **Subdivision_ID** — идентификатор корневого раздела форума, к которому относится категория;
- **Name** — название категории;
- **Description** — описание категории;
- **Priority** — приоритет категории для сортировки.

Forum_Subdivisions

В этой таблице хранятся данные соответствия корневых разделов форумов и самих форумов их разделам в системе. Таблица небольшая, но весьма информативная. Данные из этой таблицы можно использовать при выводе всех разделов форума.

```
CREATE TABLE `Forum_Subdivisions` (
  `ID` int(10) unsigned NOT NULL auto_increment,
  `Subdivision_ID` int(10) unsigned NOT NULL default '0',
  `Sub_Class_ID` int(10) unsigned NOT NULL default '0',
  `Type` enum('parent','forum') NOT NULL default 'parent',
  `Group_ID` int(10) unsigned NOT NULL,
  `Description` text NOT NULL,
  PRIMARY KEY (`ID`),
  KEY `Subdivision_ID` (`Subdivision_ID`),
  KEY `Sub_Class_ID` (`Sub_Class_ID`),
  KEY `Group_ID` (`Group_ID`)
) ENGINE=MyISAM DEFAULT CHARSET=cp1251;
```

- **Subdivision_ID** — идентификатор раздела форума;
- **Sub_Class_ID** — идентификатор компонента в разделе, основного для форума (в данном случае «Топики»);
- **Type** — тип раздела:
 - 1 *parent* — корневой раздел с форумами;
 - 2 *forum* — раздел форума.
- **Group_ID** — идентификатор группы форума;
- **Description** — описание форума.

Forum_Topics

В этой таблице хранятся данные топиков о количестве ответов на них и идентификаторы последних ответов.

```
CREATE TABLE `Forum_Topics` (  
  `ID` int(10) unsigned NOT NULL auto_increment,  
  `Subdivision_ID` int(10) unsigned NOT NULL default '0',  
  `Topic_ID` int(10) unsigned NOT NULL default '0',  
  `Replies` int(10) unsigned NOT NULL default '0',  
  `Last_Reply_ID` int(10) unsigned NOT NULL default '0',  
  PRIMARY KEY (`ID`),  
  KEY `Subdivision_ID` (`Subdivision_ID`),  
  KEY `Topic_ID` (`Topic_ID`),  
  KEY `Last_Reply_ID` (`Last_Reply_ID`)  
) ENGINE=MyISAM DEFAULT CHARSET=cp1251;
```

- Subdivision_ID — идентификатор раздела содержащего топик;
- Topic_ID — идентификатор топика;
- Replies — количество ответов на топик;
- Last_Reply_ID — идентификатор последнего добавленного ответа.

Административная часть



■ Настройки форума

В панели управления модуля ("Настройки" — "Список модулей" — "Форум") можно управлять отображением форумов на сайте, создавать и управлять категориями, изменять описания форумов и т.п.

Список форумов

Все доступные в системе форумы можно просмотреть в виде древа. Изначально отображаются только корневые разделы, которые можно раскрывать, нажатием на иконку «+».

Данный интерфейс выполнен с использованием технологии AJAX и позволяет перемещать форумы из одной категории в другую при помощи мыши, достаточно просто перетащить форум в нужную категорию в пределах корневого раздела.

При нажатии на название форума (как корневого, так и обычного), на экране отобразится поле с описанием, которое можно изменить по своему усмотрению. При клике на любом месте вне этого поля данные автоматически сохраняются.

Управление категориями

В этом разделе можно управлять доступными в системе категориями. Изменять или удалять существующие, а так же создавать новые. Изменения вносятся в систему после нажатия кнопки «Сохранить», в нижнем правом углу экрана.

■ Конвертер из старых версий

Диалог выбора

Конвертер позволяет перенести данные из старой версии форума в новую. Все действия выполняются последовательно и не должны вызвать затруднений.

На первом этапе следует выбрать сайт, на котором находится «старый» форум. После этого нужно нажать кнопку «Выбрать сайт». Конвертер отобразит все доступные для конвертирования форумы на сайте.

Далее следует выбрать интересующий форум из выпадающего списка. *Список генерируется на основе доступности в разделе прикрепленного компонента `LIST_CATEGORIES_TABLE` в «старой» версии форума.*

В этом же диалоге можно ввести желаемое ключевое слово и название нового форума. Они будут использованы при создании раздела. *Если в процессе конвертации возникнет коллизия — ключевое слово будет дополнено числовыми символами.*

Настройка прав

В этом разделе можно управлять конвертированием прав в новый форум. Это актуально, т.к. в новом форуме используются права системы, что значительно расширяет его функциональность.

Для конвертирования доступны права на форумы (разделы) и права для групп пользователей, которые будут автоматически созданы в процессе конвертации.

После нажатия кнопки «Конвертировать», данные будут перенесены на новый функционал.

○ Константы модуля

Важно: данные настройки предназначены только для опытных пользователей, установка значений констант отличных от значений по-умолчанию может привести к некорректной работе модуля в целом.

В административном списке предопределённых констант модуля ("Настройки"—"Настройка модуля"—"Форумv2"—"Настройки")

по-умолчанию прописаны следующие обязательные константы:

● **FORUM_CLASS_ID** — Идентификатор компонента "Список форумов". Значение автоматически прописывается при установке модуля или же при установке системы с уже интегрированным модулем.

● **TOPIC_CLASS_ID** — Идентификатор компонента "Топики". Значение автоматически прописывается при установке модуля или же при установке системы с уже интегрированным модулем.

● **REPLY_CLASS_ID** — Идентификатор компонента "Ответы". Значение автоматически прописывается при установке модуля или же при установке системы с уже интегрированным модулем.

● **FORUM_RSS_CLASS_ID** — Идентификатор шаблона компонента "Список форумов RSS".

● **TOPIC_RSS_CLASS_ID** — Идентификатор шаблона компонента "Топики RSS".

● **REPLY_RSS_CLASS_ID** — Идентификатор шаблона компонента "Ответы RSS".

● **FORUM_AVATAR_FIELD** — Ключевое слово поля аватары пользователя в системной таблице "Пользователи".

● **FORUM_SIGNATURE_FIELD** — Ключевое слово поля подписи пользователя, в системной таблице "Пользователи".

•**SUBSCRIBER_MAILER_ID** — Идентификатор рассылки для форума.

•**TRACKING_MODE** — Включить режим создания форума в карте сайта. В этом режиме при добавлении раздела в систему с прикрепленным компонентом "Список форумов" или "Топики" будут созданы записи в таблицах форума.

• Минимагазин

○ Возможности и требования

Модуль «Минимагазин» предназначен для организации магазина у себя на сайте, включающий:

- каталоги товаров (в том числе и неформализованные)
- корзину
- заказы (с сохранением истории)
- скидки
- уведомления по почте

Для работы модуля необходимо иметь следующие расширения PHP:

- json (<http://php.net/manual/en/book.json.php>)

Так же нужна js-библиотека jquery (<http://jquery.com/>), которая поставляется с NetCat.

○ Подключение модуля

Модуль работает с неформализованным каталогом товара, то есть товары создаются «динамически», достаточно указать наименование товара, его цену и необязательно ссылку на описание товара. Для вывода элемента «Положить в корзину» есть метод `show_put_button` глобального объекта `$nc_minishop`. Ему и передается наименование товара, цена и ссылка на описание, например, в макете дизайна вы можете написать:

```
".$nc_minishop->show_put_button('Товар1', 5000,
'/goods/tovar.html')." "
```

После чего появится товар с именем «Товар1» и ценой 5000, который можно будет положить в корзину.

Конечно, вручную задавать каждый товар не очень удобно, поэтому можно создать каталог товара на основе компонента. Пусть у нас в компоненте есть поля:

- Name – наименование товара
- Price – цена товара

Тогда в «объекте в списке» данного компонента будут доступны переменные `$f_Name` и `$f_Price` (подробнее — см. Руководство

разработчика), которые содержат наименование товара и его цену, тогда можно написать в «объекте в списке»:

```
".$nc_minishop->show_put_button($f_Name, $f_Price, $fullLink)."
```

После чего каждый объект этого компонента станет товаром в минимагазине, а значит его можно будет положить в корзину и приобрести.

Поскольку товары задаются динамически, необязательно чтобы каждый объект компонента соответствовал одному товару. Допустим, у вас есть каталог товаров, причем каждый товар может иметь несколько цветов (товары одного наименования, но различных цветов не отличаются по цене), чтобы не создавать для каждого товара несколько объектов, которые отличаются только свойством цвета, мы будем на основе одного объекта создавать несколько товаров в магазине, в «объекте в списке» компонента с товарами можно написать:

```
Красный: ".$nc_minishop->show_put_button($f_Name. 'красный',  
$f_Price, $fullLink)."  
Синий: ".$nc_minishop->show_put_button($f_Name. 'синий',  
$f_Price, $fullLink)."  
Зеленый: ".$nc_minishop->show_put_button($f_Name. 'зеленый',  
$f_Price, $fullLink)."
```

Тогда у каждого товара появится три кнопочки положить в корзину с тремя вариантами цвета.

Если товар уже в корзине, то вместо элемента «Положить в корзину» появится информер о том, что товар уже в корзине.

Шаблон отображения этих элементов задаются в настройках модуля.

■ Массовое добавление товаров в корзину

Модуль позволяет положить в корзину сразу несколько различных товаров в разном количестве. Допустим у нас есть компонент с полями Price и Name и мы хотим дать возможность пользователям положить сразу несколько товаров в корзину. Для этого в префиксе напишем:

```
". $nc_minishop->mass_put_header() ."
```

Этот код «откроет» область массового положения товаров в корзину.

В объекте в списке будем выводить поле ввода для количества для каждого товара:

```
".$nc_minishop->show_put_button($f_Title, $f_Price, $fullLink, $f_RowID) ."
```

Как вы видите, используется та же функция, что и для одиночного положения товара в корзину, за исключением того, что передается четвертый параметр — уникальный номер товара, у нас это может быть ID товара.

И, наконец, в суффиксе выводим кнопку «Положить товары в корзину»:

```
".$nc_minishop->mass_put_footer() ."
```

○ Заказы

После набора товаров в корзину, пользователь может оформить заказ. В зависимости от настроек модуля, это может сделать как неавторизированный так и только авторизированные пользователи.

По умолчанию, каждый заказ содержит в себе:

- информацию о покупателе (имя, адрес, телефон, email)
- состав заказа (позиции заказа)
- статус заказа
- способ доставки и оплаты
- скидку
- стоимость со скидкой и без нее
- примечание

С точки зрения NetCat, каждый заказ — это объект компонента «Минимагазин: Заказы», поэтому вывод, добавление и изменение заказов идет через стандартные средства системы.

Администратор сайта может просматривать все заказы, а так же изменять его, в том числе:

- добавлять, изменять, удалять позиции товара
- вручную устанавливать скидку и окончательную стоимость заказа
- править все свойства заказа

○ Скидки

Модуль позволяет задать скидку в зависимости от всей суммы заказа.

Скидка задается в административной части модуля, каждая скидка должна включать ценовой диапазон, к которому она будет применена и собственно значение скидки.

Минимагазин

Информация | **Настройки**

Общие | **Скидки** | Отображение | Шаблоны писем | Системные

☒ Предоставлять скидки в зависимости от суммы заказа

Сумма заказа от до — скидка % удалить

Сумма заказа от до — скидка % удалить

Сумма заказа от до — скидка % удалить

Добавить

Сохранить

○ Корзина

Для вывода состояния корзины используется метод `show_cart_state`. В зависимости от наличия в корзине товаров он выводит либо информер пустой или информер непустой корзины. Шаблон отображения задается в настройках модуля. Целесообразно разместить информер корзины в макете дизайна:

```
".$nc_minishop->show_cart_state()."
```

Так же для работы с корзиной доступны следующие методы:

cart_count ()

Возвращает количество товаров к корзине. Пример, выводить информер корзины только если в ней что-то есть:

```
".( $nc_minishop-> cart_count() ? $nc_minishop->show_cart_state() : "" )."
```

cart_content()

Возвращает содержимое корзины в виде массива. Пример:

```
print_r ( $nc_minishop->cart_content() );
```

cart_sum(\$with_discount = 0)

Возвращает сумму товаров в корзине. Если второй параметр будет равен «1», то скидки учитываться не будут.

Пример:

```
Сумма ".$nc_minishop->cart_sum()."
```

Обратите внимание, что для вывода состояния корзины удобнее использовать метод **show_cart_state**

cart_url ()

Метод возвращает ссылку на корзину.

Пример:

```
<a href=" ".$nc_minishop->cart_url().">Перейти в корзину</a>
```

Сама корзина выводится через компонент «Минимагазин: Корзина». В этот компоненте Вы можете править внешний вид корзины.

○ Шаблоны отображения

Все шаблоны настраиваются в административной части модуля: «Настройки» - «Отображение». Шаблоны обрабатываются функцией eval, поэтому спецсимволы необходимо экранировать.

▪ Элемент «положить в корзину»

Данный элемент уже реализован следующими способами:

- текстовая ссылка
- картинка
- текст с картинкой
- кнопка
- кнопка с полем количества

Так же вы можете задать свой шаблон отображения

В ней доступны переменные:

NetCat 4.5 Руководство по модулям

\$name — наименование товара
\$price — цена товара
\$uri — ссылка на описание товара
\$hash — хэш товара (должен передаваться)

При ajax-добавление товара этот блок должен быть обрaмлен элементом с id равным 'nc_mscont_\'\$hash' и классом 'nc_msput'

▪ Элемент «товар в корзине»

Данный элемент уже реализован следующими способами:

- текстовая надпись
- картинка

Вы можете задать свой шаблон отображения.

▪ Информер непустой корзины

По умолчанию, данный шаблон выводит сообщение в следующем формате:

В вашей корзине 3 товара на сумму 1000 руб.
Оформить заказ

Вы можете сами задать шаблон отображения этого элемента, используя следующие переменные:

- \$carturl — ссылка на корзину
- \$cartcount — количество различных товаров в корзине
- \$cartsum — общая сумма заказа
- \$cartcurrency — валюта магазина
- \$orderurl — ссылка на оформление заказа

▪ Информер пустой корзины

По умолчанию, выводит:

Ваша корзина пуста

При желании, вы можете задать другой шаблон.

▪ Уведомление о помещении товара в корзину

Данный текст выводится в диалоговом окне или всплывающем слое (в зависимости от настроек модуля) при помещении товара в корзину. Для собственной настройки шаблона Вам доступны следующие переменные:

- \$carturl -ссылка на корзину
- \$orderurl — ссылка на оформление заказа

○ **Оповещение по почте**

В настройках модуля можно включить отправку писем покупателю при добавлении нового заказа. Администратору письмо шлется автоматически. Письма отсылаются в «действии после добавления» компонента «Минимагазин: Заказы».

Шаблоны писем задаются в настройках модуля: «Настройки» - «Шаблоны писем». В них доступны следующие макропеременные:

- %SHOP_NAME — название магазина
- %USER_NAME — имя пользователя
- %ORDER_NUM - номер заказа
- %SITE_URL — адрес сайта
- %CONTENT — содержимое корзины
- %DISCOUNT — скидка
- %FINAL_COST — итоговая стоимость.