

Содержание

Об авторе	10
О рецензенте	10
Предисловие	11
О чем эта книга	11
Что нужно для чтения этой книги	12
Для кого предназначена эта книга	13
Соглашения	13
От издательства	13
Глава 1. Введение в Drupal	15
Обзор Drupal	16
Как появился Drupal	17
Что предлагает Drupal	17
Применение Drupal	18
Создание Drupal-сайта	20
Планирование сайта	20
Анализ предложенного решения	21
Сообщество Drupal	22
Вкладка Documentation (Документация)	23
Вкладка Download (Загрузка)	24
Вкладка Support (Поддержка)	25
Вкладка Forum (Форум)	26
Вкладка Contribute (Участие)	28
Вкладка Contact (Связь)	29
Лицензия Drupal	30
Резюме	31
Глава 2. Создание среды разработки	32
Среда Drupal	33
Получение и инсталляция PHP, Apache и MySQL	34
Получение и инсталляция Drupal	37
Устранение часто встречающихся проблем	42
Состояние Drupal после инсталляции	45
Обзор с точки зрения администратора	47
Создание простой Drupal-страницы	48
Резюме	51
Глава 3. Основные функции	52
Добавление модулей	52
Сторонние модули	54
Загрузка модулей	54
Инсталляция модулей	55
Конфигурирование модулей	57
Модуль Forum (Форум)	58

Модуль Comment (Комментарий)	61
Модуль Search (Поиск)	64
Работа с блоками	65
Добавление блоков	66
Конфигурирование блоков	68
Меню, основные и дополнительные ссылки	69
Резюме	76
Глава 4. Конфигурирование сайта	77
Прежде чем начать	78
Чистые ссылки	79
Сообщения об ошибках	80
Файловая система и выгрузка файлов	82
Журналы и отчеты	85
Информация о сайте	87
Обслуживание сайта	88
Резюме	89
Глава 5. Управление доступом	90
Планирование политики доступа	90
Роли	92
Права доступа	94
Пользователи	96
Администрирование пользователей	97
Параметры регистрации	99
Правила доступа	103
Резюме	106
Глава 6. Простой контент	107
Обзор контента	107
Типы контента	110
Работа с контентом	112
Работа с типами контента	112
Создание контента	114
Администрирование контента	118
Модули работы с контентом	121
Сборщик RSS	121
Подшивка	129
Резюме	133
Глава 7. Сложный контент	134
Таксономия	134
Что это и зачем?	135
Реализация управляемых таксономий в Drupal	136
Реализация тезаурусов в Drupal (теги)	145
Конструктор контента (Content Construction Kit – CCK)	146
Инсталляция CCK	147
Работа с CCK	148
Экспорт и импорт	159

Добавление полей, разработанных другими	160
HTML, PHP и отправка контента	162
Форматы ввода и фильтры	163
HTML	168
Создание страницы с дополнительными эффектами	171
Отправка страницы с дополнительными эффектами	175
Резюме	176
Глава 8. Интерфейс Drupal	177
Планирование веб-интерфейса	177
Визуальное проектирование	178
Язык	178
Изображения	179
CSS	181
Темы оформления	182
Внутренняя структура тем оформления	183
Выбор базовой темы оформления	186
Настройка тем	189
Изменение тем оформления	190
Резюме	200
Глава 9. Дополнительные возможности и модификации	201
OpenID	202
Действия и триггеры	205
Языковая поддержка	207
Локализация	208
Перевод контента	215
Производительность	220
Кэширование	220
Регулирование нагрузки	223
Динамические возможности JavaScript	225
jQuery	226
Бегущая строка с новостями	230
Резюме	233
Глава 10. Управление веб-сайтом	234
Резервное копирование	235
phpMyAdmin	235
Утилита mysqldump	238
Сценарий stop и планирование задач	239
Команда crontab	239
Выполнение задач по расписанию в Windows	241
Обслуживание веб-сайта	242
Path и Pathauto	242
Удаление модулей	247
Сопровождение пользователей	248
Поисковая оптимизация и продвижение веб-сайта	249

Модернизации	251
Обновление Drupal	252
Обновление модулей	253
Резюме	255
Приложение А. Развертывание	256
Подготовка к развертыванию	256
Подготовка хоста	258
Подготовка файлов	259
Подготовка базы данных	260
Пересылка файлов	261
Создание сайта	262
Создание файлов	262
Заполнение базы данных	263
Настройка сайта	263
Проблемы с доступом	264
Тестирование	265
Резюме	266
Предметный указатель	267

4

Конфигурирование сайта

Наиболее распространенное заблуждение, характерное для новичков – предположение, что *основы* освоить легко и, значит, они не требуют мыслительных усилий. Но в действительности все не так просто. Базовое формирование сайта обычно выполняется без особого труда, но вначале нужно осознать, *что* вы хотите реализовать, и *как* вы хотите это сделать. Формулирование того, что вы хотите от своего сайта, особенно важно именно по этой причине, так что мы обсудим эту тему здесь, но вполне возможно, что вам придется не раз к ней возвращаться.

Значит ли это, что не стоит приступать непосредственно к работе над сайтом, если вы не вполне представляете себе, чего хотите? Вообще-то нет: как и в большинстве занятий, здесь также нужно уловить момент, когда вы уже достаточно готовы приступить к разработке Dgupal-сайта. Ведь практически невозможно точно определить заранее, что потребуется для сайта и как будет обеспечена его функциональность, пока вы не поработаете с ним какое-то время. Часто приходится менять поведение сайта по заявкам пользователей.



Нужно хорошо представлять себе цель своей работы, даже если вы и не вполне уверены в средствах!

Мы определим в данной главе следующие темы для обсуждения.

- Чистые ссылки.
- Сообщения об ошибках.
- Файловая система и выгрузка файлов.
- Журналы и отчеты.
- Информация о сайте.
- Обслуживание сайта.

Здесь будет рассмотрено не все, что доступно в разделе Настройка сайта системы Dgupal. Например, настройки производительности будут подробно рассмотрены в отдельном разделе далее в книге, а некоторые темы слишком просты и не заслуживают особого внимания. Но мы рекомендуем хотя бы просмотреть все пункты этого раздела, чтобы узнать, что имеется в наличии, даже если мы и не будем рассматривать их.

Итак, предположим, что вы серьезно продумали функционирование сайта и способ его использования – тогда после базовых настроек число изменений на ходу будет минимальным.

Прежде чем начать

Прежде чем приступить к конкретной работе, имеет смысл отметить следующий важный момент. После каждого изменения настроек обязательно проверьте, как это повлияло на Drupal-сайт. Нередко бывает так, что после изменения параметров или добавления возможностей они не работают в точности так, как ожидалось. И вместо воплощения задуманного можно получить несколько не то.

Возможно, некоторые из вас не в восторге от перспективы перебирать параметр за параметром, чтобы получить нужный результат. Ну да, наверно, создание броской темы — более увлекательное занятие, но без опробования всех доступных настроек вы не станете компетентным администратором Drupal.



Вообще-то настройка конфигурации сайта похожа на закладку фундамента для создания броской темы, т.к. она снижает объем последующей работы.

Войдите в свой сайт как администратор и выберите пункт меню Управление⇒ Настройка сайта. Должна появиться примерно такая страница, как показано на рис. 4.1.

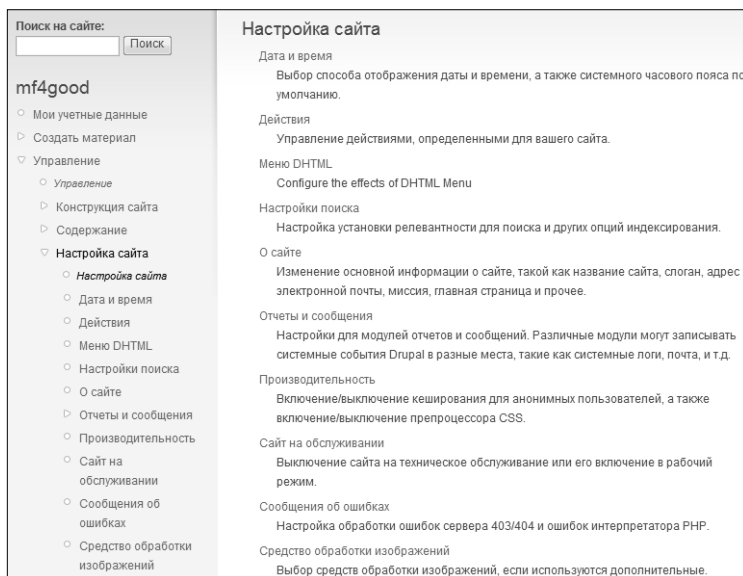


Рис. 4.1. Страница Настройка сайта

Здесь доступно множество параметров (это видно по большому количеству ссылок на этой странице), и мы в данной главе начнем потихоньку разбираться со ссылками в правой части страницы — за исключением тех, которые будут рассматриваться отдельно в последующих главах.

Глядя на последний рисунок, можно сразу увидеть, что здесь имеется ссылка Меню DHTML. Она появилась *после* того, как мы установили модуль. Это хороший пример того, как изменение в одной части сайта приводит к изменениям в другой. В данном случае ничего существенного не произошло, т.к. щелкнув на этой ссылке, мы попадем на ту же страницу конфигурации, которая доступна из представления По модулям раздела Управление.

Чистые ссылки

Важно пораньше рассмотреть эту тему, поскольку она играет роль одной из шестеренок не только в большом механизме вашего сайта, но и во взаимодействии сайта со всей всемирной Сетью. Чистые ссылки – настолько простая вещь, что трудно поверить в их важность (рис. 4.2).

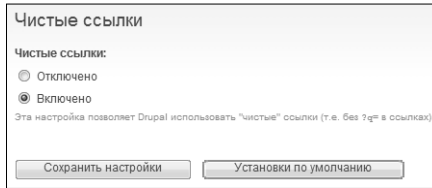


Рис. 4.2. Раздел настройки чистых ссылок

Как видите, выбор небогат: Чистые ссылки либо включены, либо выключены. Ваша система должна также сообщить, можно ли использовать чистые ссылки. Если появилось примерно такое сообщение, как на рис. 4.3, значит, возникли проблемы.

Your system configuration does not currently support this feature. The handbook page on Clean URLs has additional troubleshooting information.

Рис. 4.3. Сообщение о том, что система не поддерживает чистые ссылки

Запомните следующую рекомендацию.



Настоятельно рекомендуется, чтобы на рабочем сайте чистые ссылки были включены.

Причина этой рекомендации состоит вот в чем. Вы, конечно, хотите, чтобы ваш сайт мог по-честному конкурировать с другими сайтами, когда Google или другой поисковый механизм будет индексировать его веб-страницы. Поисковые механизмы используют автоматические программы для просмотра содержимого Сети (называемые *ботами*), и когда эти программы встречают прямые и ясные URL-адреса наподобие выводимых Drupal при включенных чистых ссылках (`http://localhost/drupal/node/2`), они спокойно выполняют индексирование страниц.

Индексирование позволяет контенту появляться в результатах поиска в Сети; тогда большее количество людей сможет находить эти страницы, и у вас все будет (более или менее) замечательно. Но если боты встречают динамические URL (которые содержат строки запроса), то они часто не так настойчиво пытаются индексировать указанные страницы, а то и вовсе игнорируют их. И получается, что у вас есть масса замечательного контента, который только и ждет, чтобы его прочитали, но никто не может прочесть его, поскольку поисковые механизмы игнорируют все ссылки вида

`http://localhost/drupal/?q=node/2`

Проблема в выделенной части этого адреса (`?q=`). Drupal перемещается по собственным страницам с помощью системы внутренних ссылок, которые имеют приведенный выше формат. То есть фрагмент `?q=node/2` указывает Drupal выбрать содержимое страницы по адресу `node/2`. Но бот Google, встретив динамический запрос, думает: “Хм... Возможно, это грязный прием, чтобы заставить меня индексировать одну и ту же страницу миллионы раз, так что я просто пропущу это место”.



Вообще-то гораздо лучше присваивать сообщениям информативные имена (называемые синонимами), чем полагаться на стандартную систему нумерации Drupal. Здесь можно забежать вперед и просмотреть раздел “Path и Pathauto” в главе 10, чтобы сразу же выработать привычку предоставлять и пользователям, и поисковым механизмам дружественные синонимы для всего вашего контента.

Разработчики Drupal знакомы с этой проблемой – поэтому, если подобное в принципе возможно, чистые ссылки по умолчанию включены, и вам вообще не о чем беспокоиться. Если вы установили пакет Apache2Triad, то ваша машина разработки безопасна в данном отношении. Но проблема может возникнуть во время развертывания, т.к. вполне возможно, что ваш поставщик Интернет-услуг не позволит использовать чистые ссылки. И что тогда?

Если вы заранее знаете, где будет находиться ваш рабочий сайт, то можете заранее выполнить проверку: установить копию Drupal на рабочем сервере и проверить, можно ли на нем использовать чистые ссылки (подробнее см. приложение А). Если нельзя, то при возможности лучше найти другой хостинг. Иначе придется связаться с системными администраторами хостинговой компании и упрашивать их до тех пор, пока они не сделают все, как надо (или просто начнут игнорировать вас).

Возможность применения чистых ссылок в основном зависит от настройки Apache. На машине разработки у вас имеется прямой доступ к файлу `httpd.conf` (находится в папке `conf` инсталляции Apache2Triad), который Apache использует для своей настройки. Но на рабочих серверах это может быть и не так, потому что любой поставщик хостинга, естественно, не желает предоставлять полный доступ к своим серверам и их настройкам.

Чтобы Drupal применял чистые ссылки, в Apache должен быть активизирован модуль `mod_rewrite`. Откройте файл `httpd.conf` и найдите в нем следующую строку:

```
LoadModule rewrite_module modules/mod_rewrite.so
```

Именно эта строка определяет, может ли Apache позволить Drupal использовать чистые ссылки. Если она закомментирована, необходимо убрать из нее комментарий, а затем перезапустить Apache, чтобы изменение вступило в силу.

Если на каком-то этапе у вас появятся проблемы, связанные с применением чистых ссылок на системе, в которой они не разрешены, то вручную зайдите на страницу

```
http://localhost/drupal/?q=admin/settings/clean-urls
```

и отключите чистые ссылки, после чего работайте с сайтом как обычно.

Не забудьте заменить выделенную часть на адрес вашего сайта.

Сообщения об ошибках

В данном разделе вы научитесь справляться с двумя распространенными ошибками, которые могут возникнуть при нормальной работе сайта. В частности, вам может понадобиться создать пару своих страниц сообщений об ошибках, которые будут выдаваться пользователям в случаях, если *страница не найдена* и *доступ запрещен*.

Вообще-то изначально существуют краткие страницы, которые выдаются по умолчанию, но если вы захотите изменить их, то процесс создания страницы с сообщением об ошибке не отличается от создания любой обычной страницы.

Давайте быстро создадим одну такую страницу. В главном меню выберите пункт Создать материал⇒Страница. Занесите любую информацию — например, сообщение *страница не найдена* (рис. 4.4).

Рис. 4.4. Создание сообщения “Страница не найдена”

Не обращайте внимания на обилие настроек, доступных на этой странице — о них мы еще поговорим ниже. А пока просто щелкните на кнопке Сохранить и запомните адрес страницы, когда она будет выведена (у меня это `http://localhost/drupal/node/4`). Теперь перейдите в раздел Сообщения об ошибках на странице Настройка сайта, занесите этот адрес в раздел Страница по умолчанию для ошибки 404 (не найдено) и щелкните на кнопке Сохранить настройки (рис. 4.5).

Рис. 4.5. Указание адресов страниц с сообщениями об ошибках

Теперь при попытке перехода на несуществующую страницу (например, `node/3333`) вы должны получить новое сообщение об ошибке, наподобие приведенного на рис. 4.6.

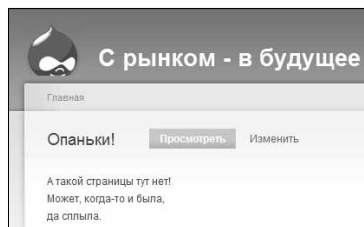


Рис. 4.6. Вывод сообщения “Страница не найдена”

В этом примере мы попросили Drupal найти узел, который не существует, и поэтому была выведена страница с сообщением об этом.

Поскольку Drupal может выдавать конфиденциальный или контент, доступный только для определенных пользователей, нам понадобится еще страница с сообщением о запрещенном доступе. На этой странице должно находиться объяснение для потенциальных пользователей, что их полномочий недостаточно для просмотра запрошенной страницы. Конечно, это не то же самое, что отсутствующий контент, но страницу Доступ запрещен можно создать точно так же.

Следующий параметр — Сообщения об ошибках — позволяет указать, выводить ли ошибки на экран и записывать в журнал или только записывать в журнал. В процессе создания сайта лучше выбрать вариант Записывать ошибки в журнал и выводить на экран, чтобы иметь возможность определить, что пошло не так и когда. Но при переводе в промышленную эксплуатацию его следует заменить (в раскрывающемся списке) на Записывать ошибки в системный журнал. Так вы не будете выдавать информацию злоумышленникам, которые могут использовать ее для атаки вашего сайта.

Регулярно просматривайте журналы в качестве составной части вашей общей стратегии, чтобы гарантировать бесперебойную работу сайта. Сообщения об ошибках, предупреждения и т.д., по сути, представляют собой окно в работу сайта и являются обязательным средством. Мы еще рассмотрим журналы и оповещения, а также принципы работы с журналами. Но пока рассмотрим файлы.

Файловая система и выгрузка файлов

Как вы будете работать с файловой системой, зависит от типа контента, который вы хотите представлять на своем сайте. Если вы заранее знаете, что все файлы будут всегда доступны для загрузки с вашего сайта любым пользователем, то оставьте для параметра Метод загрузки (на странице Управление⇒Настройка сайта⇒Файловая система) значение Публичный. Между значениями Публичный и Приватный можно переключаться, но тогда возможно появление проблем — из-за того, что при таких переключениях будут изменяться все пути файлов.



Пожалуй, лучше оставить для параметра Метод загрузки значение Публичный, т.к. в настоящее время вариант Приватный может помешать правильной работе других частей сайта.

К публичным файлам можно обратиться непосредственно из браузера, не выполняя специальных действий на вашем Drupal-сайте. Так что, если кому-то понравится размещенный на вашем сайте видеоклип, он может поместить на своем веб-сайте ссылку на него, и пользователи будут заходить на его сайт, чтобы посмотреть клип — но за счет пропускной способности вашей сети! Конечно, это не лучший вариант. Необходимо отслеживать такие моменты и в случае возникновения проблем узнать, предоставляет ли служба вашего хостинга какой-нибудь вариант защиты ссылок.

А если вы все же хотите сделать метод загрузки приватным, вам необходимо вынести каталог files из корневого каталога, чтобы он не был непосредственно доступен через Сеть. В таком случае для файлов нужно будет указывать не *относительный*, а *полностью определенный* путь, т.к. иначе Drupal будет искать их в обычном месте. Например, на машине разработки демонстрационного сайта выбран метод загрузки Публичный и следующий путь файловой системы:

```
C:\Apache2Triad\files
```

Кроме того, нужно, чтобы был правильно задан каталог temp; в нашем случае это

```
C:\Apache2Triad\temp
```

А теперь проверим, возможно ли без проблем загрузить файл на сайт. Перейдите на страницу Управление⇒Конструкция сайта⇒Модули, включите модуль Upload (если он еще не включен) и щелкните на кнопке Сохранить настройки. Теперь на странице Создать Страница появится параметр Прикрепленные файлы (рис. 4.7).

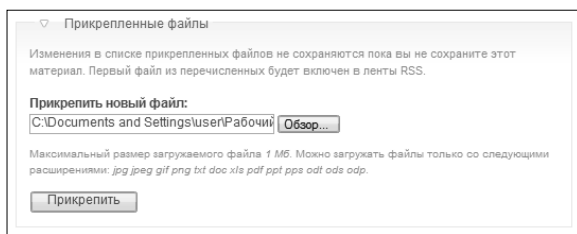


Рис. 4.7. Поле для прикрепления файла

После загрузки файла появится подтверждение (рис. 4.8).

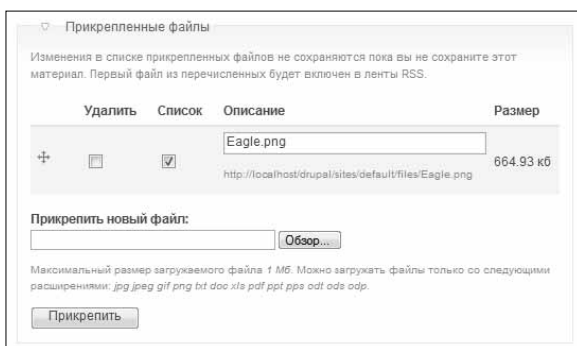


Рис. 4.8. Список прикрепленных файлов

Теперь здесь присутствует таблица, состоящая из трех столбцов:

- Удалить – позволяет удалить прикрепленный файл из сообщения;
- Список – выводит загруженный файл в разделе прикрепленных файлов данного сообщения;
- Описание – имя (и местоположение) файла.

Если вы прикрепляете файлы, чтобы выводить их на странице, то они не должны быть слишком большими, иначе это снизит скорость и производительность сайта. Кстати, если флажок Список установлен, то статья будет выглядеть так, как показано на рис. 4.9.

Для загрузки файла пользователи должны щелкнуть на ссылке. Но если вы прикрепляете файл к сообщению, чтобы он выводился на странице, то отметку с флажка Список нужно снять. Более подробно о прикрепленных файлах написано в главе 7.

Управление типами и размерами загружаемых файлов сосредоточено в разделе Управление⇒Настройка сайта⇒Загрузки файлов. Не стоит разрешать загружать на сайт файлы любых типов. Ведь тогда кто-то сможет загрузить вредоносный исполняемый файл, который будет выполнять нехорошие действия на машинах пользователей, а те, в свою очередь, будут говорить нехорошие вещи вам.

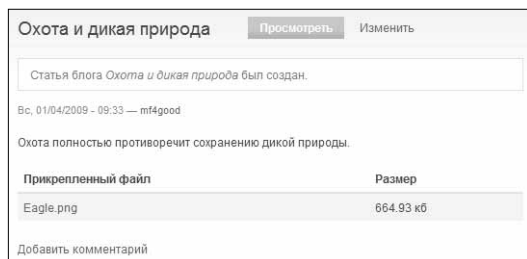


Рис. 4.9. Список прикрепленных к статье файлов

Внешний вид страницы Загрузки файлов показан на рис. 4.10.

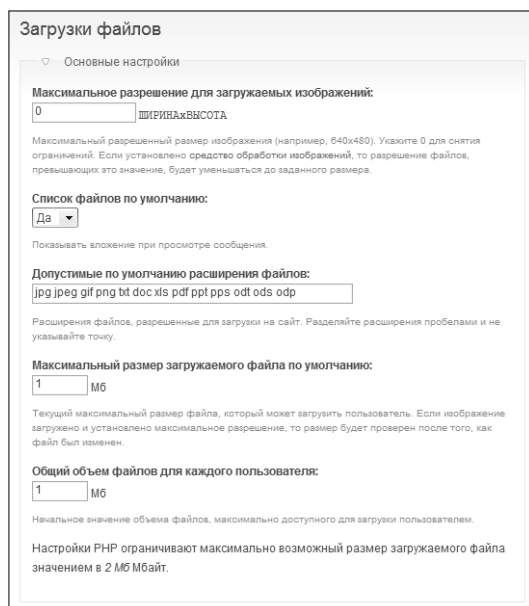


Рис. 4.10. Внешний вид страницы Загрузки файлов

На ней имеется множество настроек, и решения по ним следует принимать, исходя из потребностей конкретного сайта. При возникновении сомнений пользуйтесь следующим правилом.



Разрешайте только то, что абсолютно необходимо, и ничего больше!

Я предлагаю удалить из списка все расширения файлов, которые *точно* не понадобятся на сайте. Учтите, что вредоносное ПО можно пронести и в других типах файлов, поэтому чем больше типов вы разрешите, тем меньшим будет уровень безопасности.

PHP поставляется со встроенным пакетом обработки изображений GD — значит, вы всегда можете изменять размеры изображений, поскольку Drupal поддерживает работу GD. А это означает, что вы можете спокойно указать максимальную ширину

и высоту изображения, зная, что большие изображения можно аккуратно сжать. Это может быть очень важным для поддержки однотипности компоновки: ведь если разрешить пользователям загружать изображения шире вашей страницы, то получится очень неприятная картина.

Следует также тщательно обдумать и общий объем файлов для каждого пользователя — вряд ли вы хотите, чтобы самые активные члены сообщества могли выгрузить на сайт лишь несколько файлов. Но и полная свобода пользователей также может привести к неприятным последствиям.

Журналы и отчеты

Выберите пункт меню Управление⇒Настройка сайта⇒Отчеты и сообщения. Если вы увидите пустую страницу, активизируйте модуль Database logging в разделе Модули и вернитесь сюда еще раз. Щелкните на ссылке Отчеты в базе данных, и вы увидите доступные настройки (рис. 4.11).

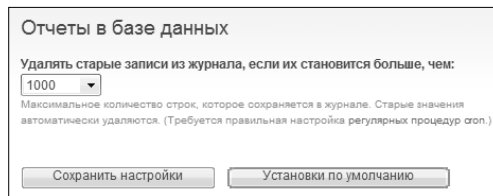


Рис. 4.11. Настройка хранения отчетов в базе данных

Настройка всего одна, но, по крайней мере, она вполне понятна: количество хранимых в системе записей можно увеличить или уменьшить в зависимости от объема работы по сопровождению сайта. Учтите, что Dgupal может вести журнал событий для сайта только при регулярном выполнении заданий cron (подробнее об этом см. в главе 10). Единственный параметр — это не очень интересно, так что взглянем на сами журналы, которые хранит Dgupal.

Для этого рассмотрим раздел Отчеты на странице Управление — там находится список различных доступных журналов и отчетов (рис. 4.12).

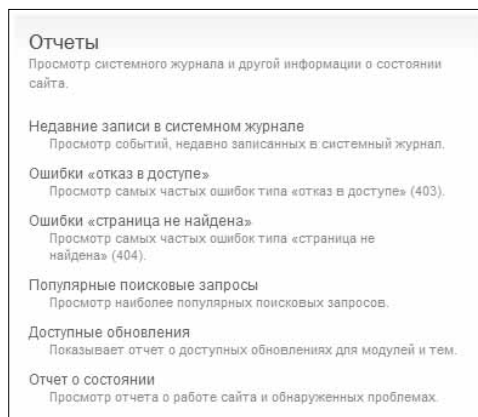


Рис. 4.12. Раздел Отчеты страницы Управление

Если щелкнуть на первой ссылке — Недавние записи в системном журнале — появится журнал событий сайта. Его записи можно фильтровать; щелчок на ссылке Фильтр системных сообщений открывает два списка Тип и Важность (рис. 4.13).

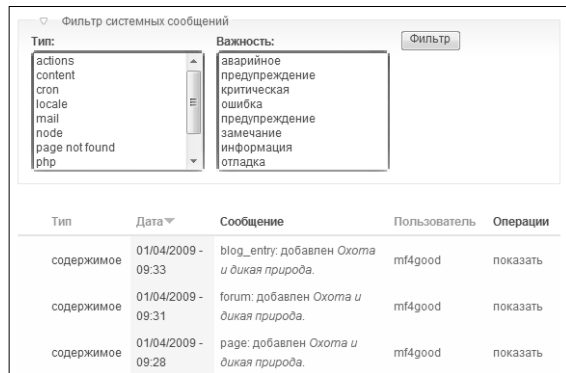


Рис. 4.13. Настройка фильтра системных сообщений

У каждой записи есть несколько важных характеристик, которые помогают определить их тип и важность, кто или что их вызвало и результат события. Если нужно просмотреть подробные сведения о каком-то сообщении об ошибке, щелкните на ссылке в столбце Сообщение, и вы увидите примерно такую информацию, как показано на рис. 4.14.

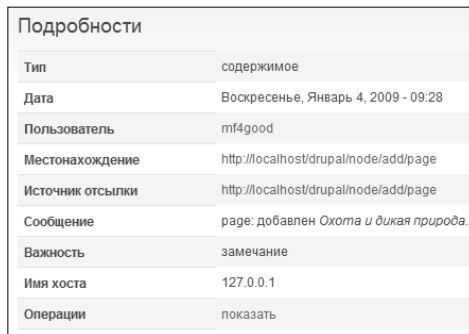


Рис. 4.14. Пример сообщения об ошибке (замечание)

Подобный интерфейс предоставляет довольно широкие возможности по обнаружению и устранению проблем с сайтом. В разделе Отчеты есть и другие пункты, которые вы можете просмотреть самостоятельно. Наиболее интересны Доступные обновления и Отчет о состоянии — особенно на момент выхода книги, т.к. многие разработчики модулей и самого Drupal еще занимаются обновлениями.

Регулярный просмотр различных отчетов может помочь обнаружить и изолировать необычные или важные события, которые существенно влияют на сайт. Вы можете обнаружить, что на каком-то сайте ряд людей разместил ссылки на ваш сайт, и, значит, это хороший кандидат на установление отношений. А возможно, хакеры и спамеры пытались получить доступ к определенным страницам, и следы их действий зафиксированы в журналах (например, их IP-адреса и т.д.).

Информация о сайте

На этой странице (Управление⇒Настройка сайта⇒О сайте) содержатся разнообразные параметры; некоторые из них вполне понятны, а другие заставляют серьезно задуматься. Все параметры представляют собой текстовые поля, которые определяют имя сайта, заявление о его главной цели и т.д. (рис. 4.15).

О сайте

Имя: *
С рынком - в будущее
Название этого сайта.

E-mail адрес: *
neverbeen@list.ru
Исходящий адрес e-mail для автоматической отправки писем при регистрации, запроса нового пароля и других уведомлений. (Используйте адрес, связанный с вашим доменом, чтобы избежать попадания писем в спам)

Слоган:
Девиз сайта

Миссия:
Миссия вашего сайта или главное, на что вы хотите обратить внимание (часто отображается на главной странице).

Сообщение внизу страницы:
Этот текст отображается внизу каждой страницы. Удобен для размещения замечания об авторских правах.

Анонимный пользователь: *
Гость
Это имя присваивается анонимным пользователям.

Главная страница по умолчанию: *
http://localhost/drupal/node
Главная страница показывает содержимое, расположенное по этому адресу. Если затруднитесь, укажите "node".

Рис. 4.15. Страница настройки информации о сайте

Ничего экстраординарного, хотя я бы посоветовал воздержаться от формулировки миссии и девиза, пока не решите, какую тему оформления вы хотите реализовать. Причина состоит в том, что различные темы обрабатывают эти параметры по-разному, а некоторые и вовсе никак.

Просматривая эти поля, можно увидеть, что, оказывается, есть возможность присвоить имя анонимным пользователям. Обычно используется имя Гость, но при наличии веских причин его можно сменить на что-нибудь другое. Все, что при этом изменится — вместо имени Гость в сообщениях будет другое имя (если анонимным пользователям разрешено отправлять сообщения на этот сайт).

Последний параметр достоин большего внимания, поскольку многие предпочитают иметь определенную стартовую страницу, с которой можно добраться до любого нужного контента — в отличие от стандартного поведения, когда последние сообщения заносятся на главную страницу.

Но может понадобиться выводить в качестве стандартной страницу с другим контентом, на которую пользователи должны попадать до того, как увидят другой контент. Например, если вы хотите показывать какую-то рекламную информацию на вводной

странице, с помощью данного параметра можно указать Drupal выводить именно эту страницу. Только, чтобы указать страницу, вначале нужно создать контент и определить его путь. Можно, к примеру, с помощью номера указать конкретный узел, а можно и блоги сайта — для этого нужно выбрать номер в формате `узел/номер`.

Определить путь головной страницы можно, просто зайдя на данную страницу (если она уже создана). Это может быть страница блога, канал агрегированных новостей (о каналах будет рассказано ниже), форум и что угодно еще.



Когда вы определитесь с контентом, предназначенным для головной страницы, узнайте ее **относительный путь** и просто введите его в текстовое поле.

Вспомните, что относительный путь — это часть адреса страницы, которая находится после стандартного домена, общего для всего сайта. Например, относительным путем может быть `node/2`, т.к. Drupal преобразует такой путь в

```
http://localhost/drupal/node/2
```

Первая часть этого адреса — `http://localhost/drupal/` — является **базовым адресом**, а все, что после него, является относительным путем.

Обязательно проверьте, что все работает как надо. Ведь при указании путей файлов легко допустить ошибку или добавить где-нибудь лишний слеш.

Обслуживание сайта

Еще раз напоминаем следующее.



Вся разработка сайта или внесение в него серьезных изменений должны выполняться на машине разработки и тщательно проверяться, прежде чем перенести результаты на рабочий сайт.

Но все же бывают случаи, когда нужно просто внести какие-то изменения непосредственно на рабочем сайте — например, реализовать обновление, тщательно проверенное на сервере разработки. В таком случае можно закрыть сайт для обслуживания, чтобы пользователи не смогли заходить на него в это время, а для этого достаточно переключить Состояние сайта в режим Обслуживание (Управление⇒Настройка сайта⇒Сайт на обслуживании).

При этом можно добавить небольшое сообщение с объяснением, почему сайт пока не доступен, чтобы при попытке войти на сайт пользователь видел что-то вроде рис. 4.16.

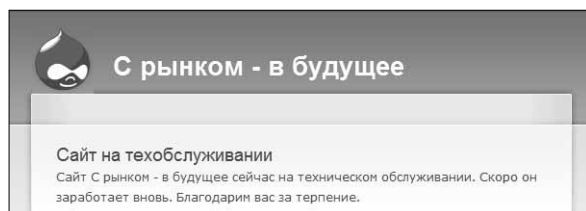


Рис. 4.16. Страница-извещение о том, что сайт находится на обслуживании

Будьте осторожны: если вы выйдете из сайта, то он будет недоступен и для вас. Это произойдет потому, что с сайтом, переведенным в автономный режим, может работать лишь User 1 (пользователь-администратор). Если вы выйдете и попытаетесь снова войти, вы будете уже не администратором, а анонимным пользователем, и увидите только такое сообщение.

И поскольку звание администратора сайта в таком случае не помогает, Drupal позволяет войти на страницу входной регистрации как на любую другую. Введите адрес `http://localhost/drupal/user`, и вы сможете войти как администратор и беспрепятственно пользоваться сайтом.

Очень важно помнить о следующем.



При переводе состояния сайта в автономный режим вы должны помнить пароль администратора!

Все остальные пользователи будут заблокированы, пока Состояние сайта не будет возвращено в состояние Онлайн.

Резюме

В данной главе было рассмотрено значительное количество материала, касающегося настройки сайта. Мы начали со знакомства с некоторыми общими конфигурационными параметрами, которые важны для приведения механизма сайта в рабочее состояние. Многие из этих параметров еще придется пересмотреть при разработке сайта и по мере освоения.

В условиях жесткой конкуренции в среде Интернет мы должны использовать все доступные преимущества SEO — в нашем случае это означает важность активизации чистых ссылок. Если вы застрянете на этом, то, возможно, лучше поискать новую службу хостинга, чтобы не иметь проблем при настройке Apache.

Добавьте к этому возможность управлять выгрузками файлов и управление файлами, а также работу с множеством событий, которые система Drupal может протоколировать — и вы с определенной долей уверенности готовы работать с Drupal.

В конце главы было кратко рассмотрено, как вносить на сайт основные виды информации и переключать сайт в режим обслуживания, если понадобится выполнить какие-то технические действия.