3 Разработка сайта для туроператора Азия Бас

**О компании**

Туроператор ООО «Агентство путешествий и экскурсий «АЗИЯ БАС-плюс» приглашает вас отдохнуть и получить незабываемые впечатления в живописных уголках нашей планеты. Таиланд, Европа, Вьетнам, Китай, Индонезия, Россия, Приморье – далеко не весь перечень, который доступен в нашей компании для выбора увлекательного путешествия.

Перечень услуг, которые предоставляет ООО «Агентство путешествий и экскурсий «АЗИЯ БАС-плюс»:

* Бронирование чартерных программ в страны Юго-Восточной Азии;
* Организация индивидуальных и групповых выездов в Китай;
* Под индивидуальные пожелания разрабатываем специальные туры;
* Свадебные кортежи;
* Корпоративные туры для организаций, компаний, фирм;
* Экскурсии по Хабаровску и краю для детей, студентов и взрослых;
* Трансфер;
* Оформление виз, бронирование ж/д и авиа-билетов, гостиниц;
* Автотуры в Приморье и самостоятельные заезды.

Девиз ООО «Азия Бас» - Хватит мечтать, пора отдыхать.

3.1 Технологии и инструменты для разработки

Для создания сайта для туроператора Азия Бас будут использоваться следующие технологии:

1. CMS Drupal 7.54;

Система управления содержимым (контентом) (англ. Content management system, CMS) — информационная система или компьютерная программа, используемая для обеспечения и организации совместного процесса создания, редактирования и управления содержимым, иначе — контентом (от английского content)

Основные функции CMS:

* + - * Предоставление инструментов для создания содержимого, организация совместной работы над содержимым,
      * Управление содержимым: хранение, контроль версий, соблюдение режима доступа, управление потоком документов и т. п.,
      * Публикация содержимого,
    - Представление информации в виде, удобном для навигации, поиска.

Drupal (Друпáл) — система управления содержимым (CMS), используемая также как каркас для веб-приложений (CMF), написанная на языке PHP и использующая в качестве хранилища данных реляционную базу данных (поддерживаются MySQL, PostgreSQL и другие[3]). Drupal является свободным программным обеспечением, защищённым лицензией GPL, и развивается усилиями энтузиастов со всего мира.

1. СУБД MySql 5.1;

Система управления базами данных (СУБД) — совокупность программных и лингвистических средств общего или специального назначения, обеспечивающих управление созданием и использованием баз данных.

MySQL – свободная реляционная система управления базами данных. Разработку и поддержку MySQL осуществляет корпорация Oracle, получившая права на торговую марку вместе с поглощённой Sun Microsystems, которая ранее приобрела шведскую компанию MySQL AB. Продукт распространяется как под GNU General Public License, так и под собственной коммерческой лицензией.

1. PHP 5.6;

PHP (англ. PHP: Hypertext Preprocessor ­ «PHP: препроцессор гипертекста»; первоначально Personal Home Page Tools — «Инструменты для создания персональных веб-страниц») — скриптовый язык общего назначения, интенсивно применяемый для разработки веб-приложений. В настоящее время поддерживается подавляющим большинством хостинг-провайдеров и является одним из лидеров среди языков, применяющихся для создания динамических веб-сайтов.

1. Apache HTTP-сервер ­ свободный веб-сервер;

Веб-сервер ­ сервер, принимающий HTTP-запросы от клиентов, обычно веб-браузеров, и выдающий им HTTP-ответы, как правило, вместе с HTML-страницей, изображением, файлом, медиа-потоком или другими данными.

1. Html5;

HTML5 (англ. HyperText Markup Language, version 5) – язык для структурирования и представления содержимого всемирной паутины. Это пятая версия HTML.

1. Less;

LESS – это динамический язык стилей. Он создан под влиянием языка стилей Sass, и, в свою очередь, оказал влияние на его новый синтаксис «SCSS», в котором также использован синтаксис, являющийся расширением СSS.

CSS (англ. Cascading Style Sheets — каскадные таблицы стилей) — формальный язык описания внешнего вида документа, написанного с использованием языка разметки.

1. JavaScript;

JavaScript – мультипарадигменный язык программирования. Поддерживает объектно-ориентированный, императивный и функциональный стили. Является реализацией языка ECMAScript (стандарт ECMA-262).

1. jQuery 1.10;

jQuery – библиотека JavaScript, фокусирующаяся на взаимодействии JavaScript и HTML. Библиотека jQuery помогает легко получать доступ к любому элементу DOM, обращаться к атрибутам и содержимому элементов DOM, манипулировать ими.

DOM (от англ. Document Object Model ­ «объектная модель документа») ­ это не зависящий от платформы и языка программный интерфейс, позволяющий программам и скриптам получить доступ к содержимому HTML-, XHTML- и XML-документов, а также изменять содержимое, структуру и оформление таких документов.

1. Yarn/npm/bower.

Yarn/npm/bower – пакетные менеджеры.

Система управления пакетами ­ набор программного обеспечения, позволяющего управлять процессом установки, удаления, настройки и обновления различных компонентов программного обеспечения.

Инструменты для разработки:

1. Adobe Photoshop Creative Cloud 2015

Adobe Photoshop ­ многофункциональный графический редактор, разработанный и распространяемый фирмой Adobe Systems. В основном работает с растровыми изображениями, однако имеет некоторые векторные инструменты.

1. Sublime Text 3

Sublime Text ­ кроссплатформенный проприетарный текстовый редактор. Поддерживает плагины на языке программирования Python.

1. Gulp

Gulp ­ таск-менеджер для автоматического выполнения часто используемых задач (например, минификация, тестирования, объединения файлов), написанный на языке программирования JavaScript. Программное обеспечение использует командную строку для запуска задач, определённых в файле Gulpfile.

1. Node JS

Node или Node.js ­ программная платформа, основанная на движке V8 (транслирующем JavaScript в машинный код), превращающая JavaScript из узкоспециализированного языка в язык общего назначения. Node.js добавляет возможность JavaScript взаимодействовать с устройствами ввода-вывода через свой API (написанный на C++), подключать другие внешние библиотеки, написанные на разных языках, обеспечивая вызовы к ним из JavaScript-кода. Node.js применяется преимущественно на сервере, выполняя роль веб-сервера.

1. Git

Git ­ распределённая система управления версиями. Проект был создан Линусом Торвальдсом для управления разработкой ядра Linux, первая версия выпущена 7 апреля 2005 года.

Система управления версиями (от англ. Version Control System, VCS или Revision Control System) ­ программное обеспечение для облегчения работы с изменяющейся информацией. Система управления версиями позволяет хранить несколько версий одного и того же документа, при необходимости возвращаться к более ранним версиям, определять, кто и когда сделал то или иное изменение, и многое другое.

1. GitHub

GitHub ­ крупнейший веб-сервис для хостинга IT-проектов и их совместной разработки. Основан на системе контроля версий Git и разработан на Ruby on Rails и Erlang компанией GitHub

1. ConEmu

ConEmu-Maximus5 это «графический хост» для консольных приложений Windows. При запуске нового консольного приложения в ConEmu реальное окно консоли не отображается, а весь вывод и ввод перехватывается и перенаправляется в ConEmu.

1. PhpStorm

JetBrains PhpStorm — коммерческая кросс-платформенная интегрированная среда разработки для PHP. Разрабатывается компанией JetBrains на основе платформы IntelliJ IDEA.

Интегрированная среда разработки (англ. Integrated Development Environment) — система программных средств, используемая программистами для разработки программного обеспечения.

1. OpenServer

Open Server Panel ­ это портативная серверная платформа и программная среда, созданная специально для веб-разработчиков с учётом их рекомендаций и пожеланий.

1. Google Chrome

Google Chrome (с англ. — «хром») — браузер, разрабатываемый компанией Google на основе свободного браузера Chromium и движка Blink.

1. Bootstrap

Bootstrap (также известен как Twitter Bootstrap) ­ свободный набор инструментов для создания сайтов и веб-приложений. Включает в себя HTML- и CSS-шаблоны оформления для типографики, веб-форм, кнопок, меток, блоков навигации и прочих компонентов веб-интерфейса, включая JavaScript-расширения.

3.2 Верстка

Вёрстка веб-страниц ­ создание структуры html-кода, размещающего элементы веб-страницы (изображения, текст и т. д.) в окне браузера, согласно разработанному макету, таким образом, чтобы элементы дизайна выглядели аналогично макету.

Вёрстка веб-страниц отличается от полиграфической тем, что необходимо учитывать разницу отображения элементов в различных браузерах и разницу в размерах рабочего пространства устройств.

Процесс сложен и имеет творческую основу, ни один из способов не является каноничным и принятым как основа. Все подходы к вёрстке имеют как преимущества, так и недостатки.

Хотя работа верстальщика скрыта от глаз, именно она обеспечивает бесперебойность при работе на различных устройствах, а также скорость загрузки каждой страницы сайта.

3.2.1 Настройка автоматизации разработки

Что требуется установить для работы:

* Node;
* Git;
* yarn;
* gulp;
* bower;
* sublime text 3

Для начала верстки, необходимо настроить автоматическую сборку, Livereload (автоматическая перезагрузка страницы), чтобы ускорить процесс разработки. Переходим в консоль ConEmu, в ней нужно эмулировать консоль GitBush и прописать следующие команды, после выполнения каждой нужно нажимать клавишу Enter, после этого можно будет настраивать проект и создавать базовую структуру проекта.

Операции в консоли:

* mkdir asiabus – создание корневой директории;
* cd !$ - переход в созданную директорию;
* mkdir app – папка для содержимого проекта;
* yarn init –y – создание файла package.json и yarn.lock
* touch gulpfile.js .gitignore .bowerrc bower.json

Далее, при помощи yarn, нам необходимо установить модули для gulp и прописать их таски (работу) в файле gulpfile.js

Вводим следующую команду:

yarn install –D gulp gulp-less path gulp-connect gulp-livereload gulp-wiredep

Этой командой у нас устанавливаются зависимости в папку node\_modules, которая создается автоматически, флаг –D указывает, что эти зависимости должны попасть в файл package.json, в массив devDependencies, то есть в зависимости для разработки.

Далее настраиваются таски этих модулей в файле gulpfile.js.

В файле .bowerrc указываем директорию, в которую будут устанавливаться, библиотеки, скачанные при помощи пакетного менеджера bower. Настройка таска в gulpfile.js (bower) и специальные комментарии в index.html будет подключать bower\_components автоматически, используя содержимое файла package.json.

Все зависимости устанавливаются в папку app/bower\_components, и записываются в массив dependencies файла bower.json.

Bower install -S [имя зависимости] (например jquery или bootstrap)

После настройки, в консоли нужно выполнить команду gulp

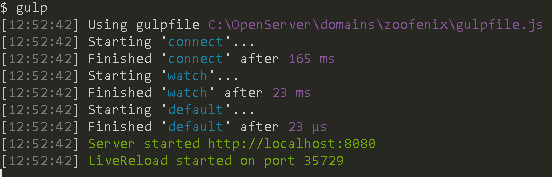


Рисунок – выполнение команды gulp

Запущен локальный сервер с адресом localhost:8080, который можно ввести в браузере, теперь можно создавать структуру приложения.

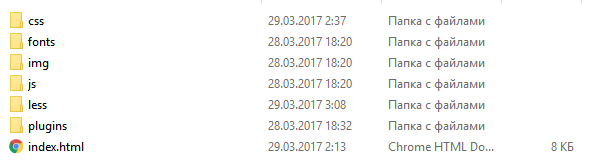


Рисунок – содержимое папки app

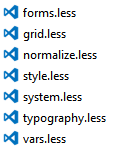


Рисунок – Содержимое папки less

Все стили будут создаваться при помощи css-препроцессора less, настройка модулей gulp-livereload и gulp-less позволяет, при каждом сохранении файла стилей, преобразовывать файлы с расширением .less в файлы с расширением .css и помещать их в папку css. В index.html достаточно подключить файл style.css. При каждом сохранении любого .js, .less, .css и .html файла, браузер автоматически обновляется (при работе с двумя мониторами это очень удобно, так как не нужно переходить в браузер и обновлять страницу после каждого сохранения файлов).

Теперь можно создать пустой git-репозиторий. В корневой папке проекта вводим следующие команды:

* git init – создаем репозиторий;
* git add all – добавляем все файлы в отслеживание;
* git commit –m “first commit” – коммитим (фиксируем) изменения в файлах.

Автоматизация настроена, проект находится под управлением системы контроля версий, можно приступать к верстке, используя psd-макет, который можно открыть в программе adobe photoshop.

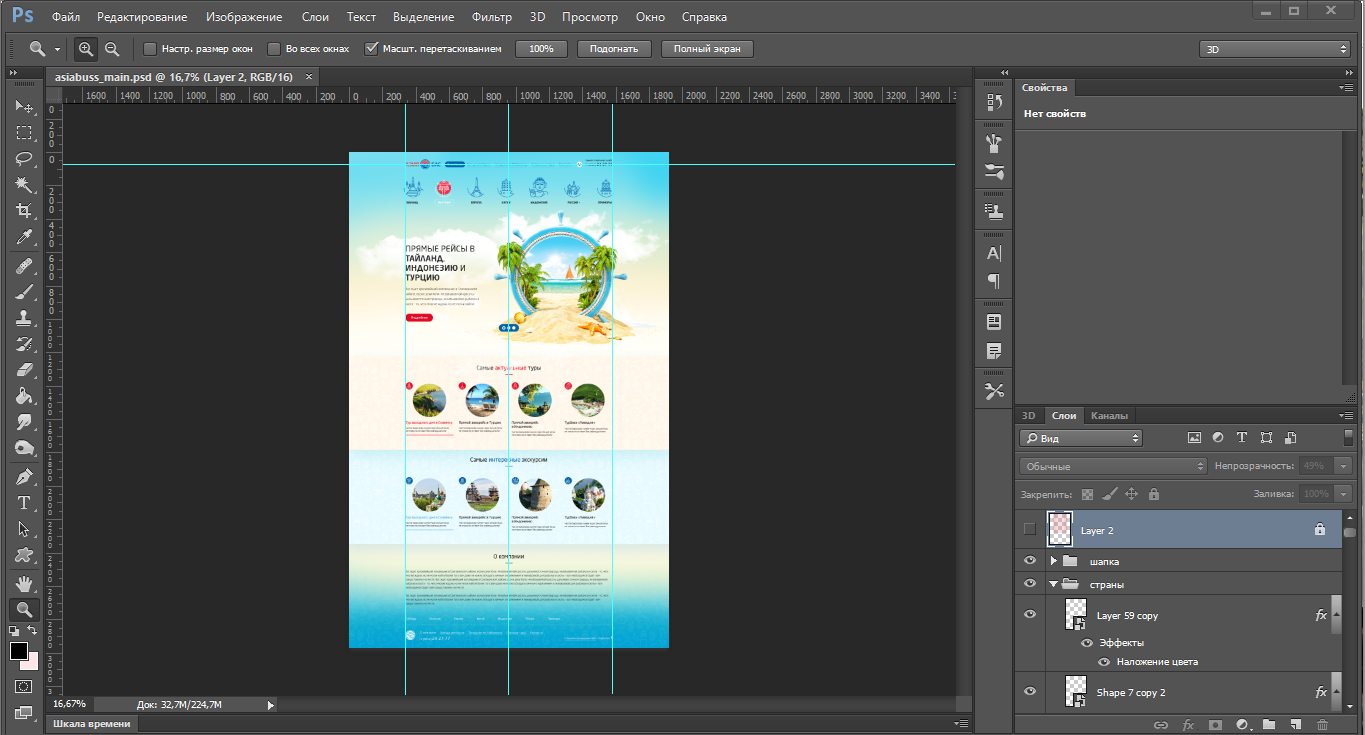


Рисунок – psd-макет в Adobe Photoshop.

Далее, в текстовом редакторе sublime text 3 пишутся все less, js и html файлы.

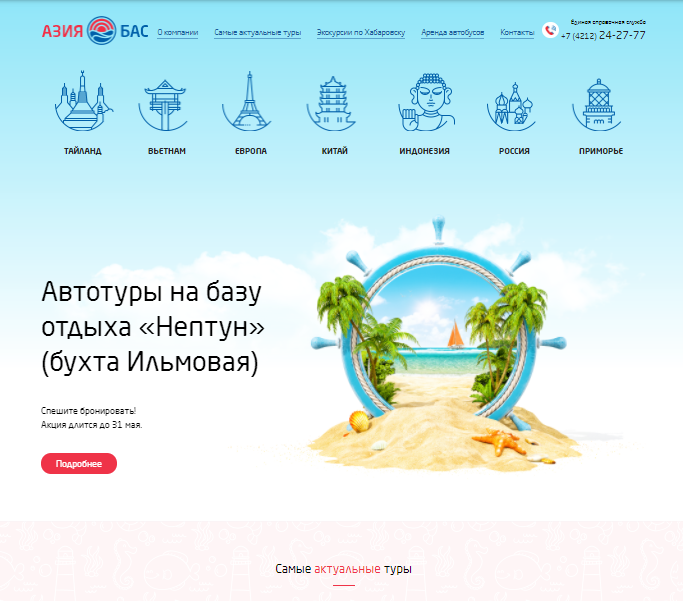


Рисунок – результат верстки

3.3 Натяжка верстки на движок Drupal

После того, как готова статичная верстка сайта, необходимо сделать сайт динамичным и добавить к нему административную часть, для удобного управления содержимым сайта. Для начала работы нужно запустить локальный сервер Open Server, запустить его и далее перейти в PhpMyAdmin.

3.3.1 Установка Drupal

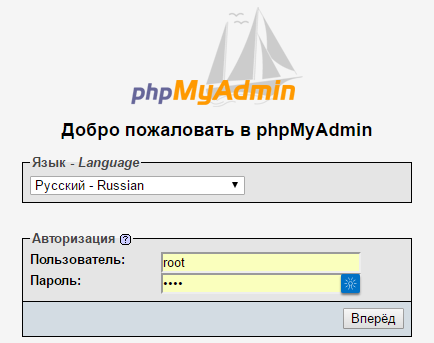


Рисунок – Окно входа в phpMyAdmin

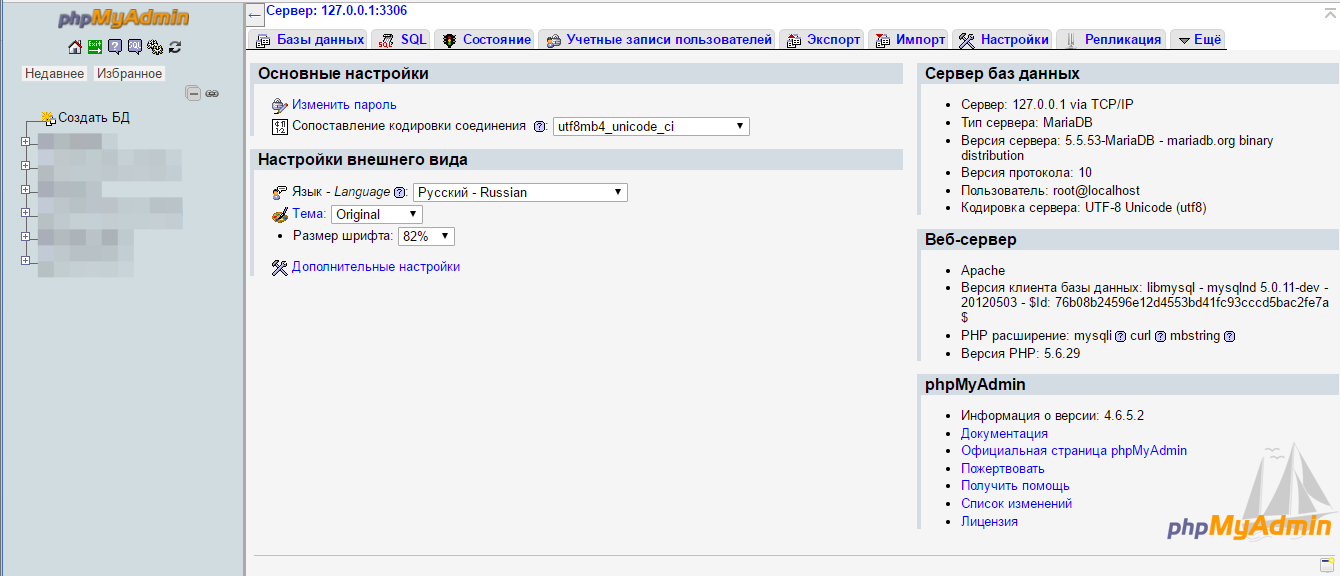


Рисунок – Главное окно phpMyAdmin

Переходим в пункт «Создать базу данных» и создаем новую БД, указав ее название и сравнение.

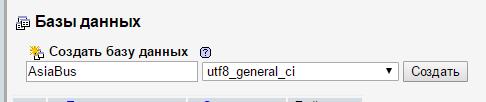


Рисунок – Создание новой БД

База данных готова, создание таблиц, связей и наполнение данными будет происходить автоматически, когда база будет подключена к сайту под управлением Drupal.

Теперь в папке domains корневого каталога Open Server нужно создать папку, в которую далее установится CMS Drupal.

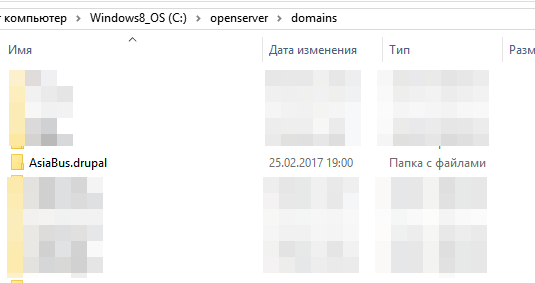


Рисунок – Папка для содержимого сайта.

В эту директорию нужно перенести все файлы из архива, в котором находится сам Drupal, после этого в меню Open Server, в пункте «Мои сайты» выбрать соответствующий, после этого откроется браузер и мы увидим окно приветствия Drupal, которое предложит нам установку.

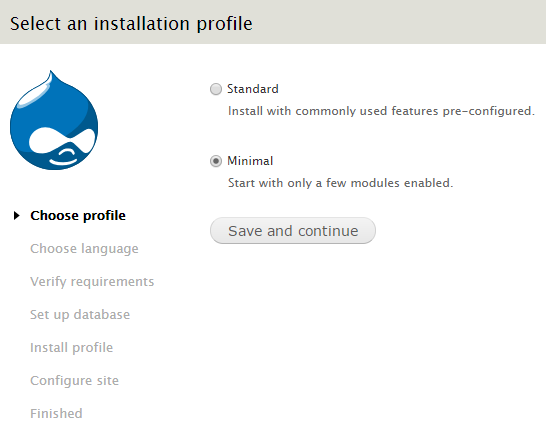


Рисунок – Выбор профиля установки

Выбираем минимальный профиль установки, так как потом будет проще включить нужные модули и функции, чем убирать лишние.

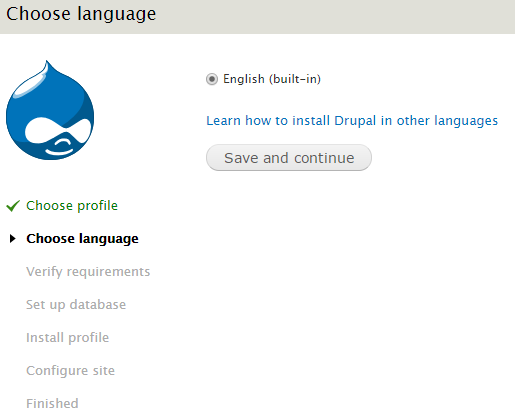


Рисунок – Выбор языка

По умолчанию в Drupal 7 только один язык для выбора, но после установки можно будет установить любой язык.

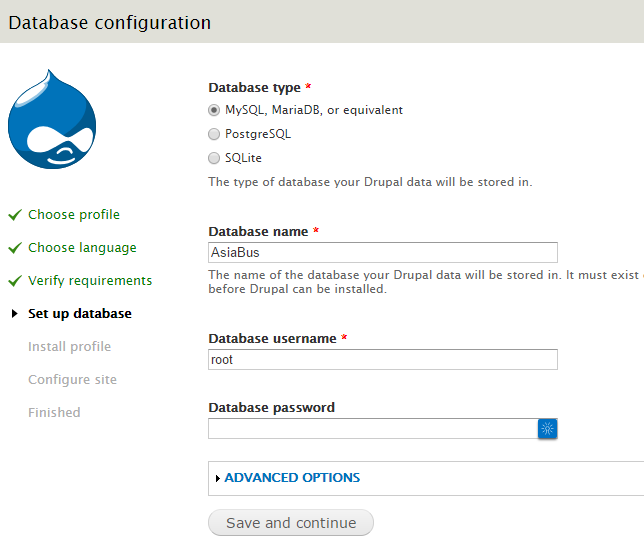


Рисунок – Подключение к бд

После идет установка минимального профиля.

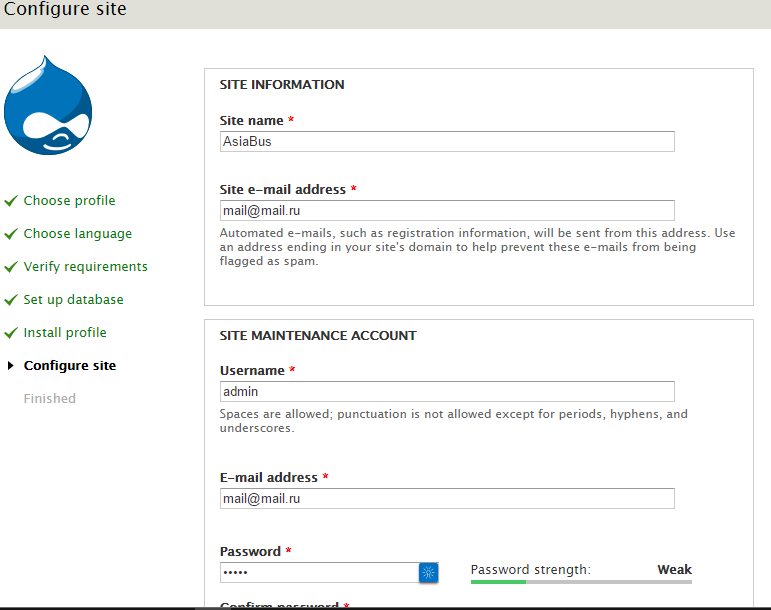


Рисунок – Конфигурация сайта

В данном окне настраивается название сайта, главный администратор сайта и т.д.

CMS Drupal 7.54 Установлена

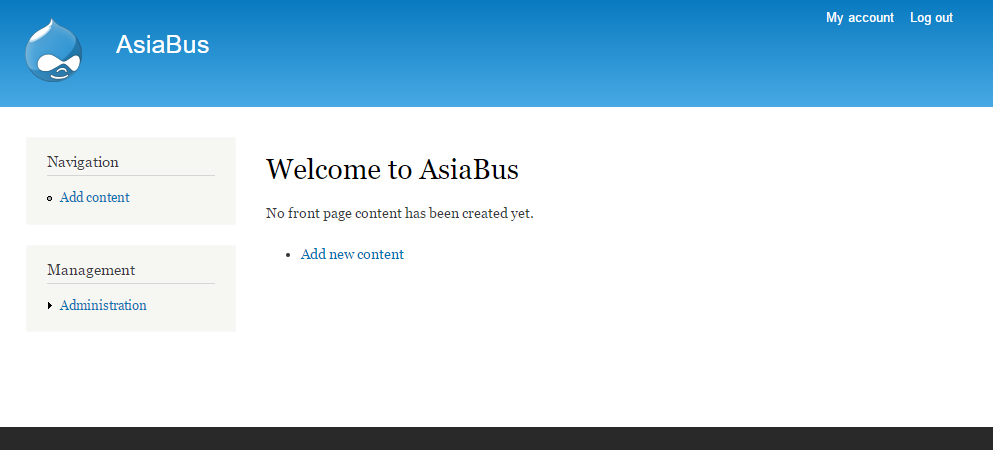


Рисунок – Главная страница сайта.

3.3.2 Первоначальная настройка и установка модулей

Все модули для разработки, интерфейса пользователя и т. д. будут скачиваться с официального сайта [www.drupal.org/project/project\_module](http://www.drupal.org/project/project_module)

Для настройки интерфейса нам понадобятся следующие модули:

1. Admin menu. Добавляет выпадающее меню для большинства административных и других общих задач (для пользователей с соответствующими правами).
2. Module filter. Фильтрует список модулей.
3. Empty Front Page. Удаление контента по умолчанию с главной страницы.
4. Localization Update. Предоставляет возможность автоматической загрузки и обновления переводов.
5. Administration menu Toolbar style. Улучшенная Панель инструментов.

Переходим пункт меню Administration, далее modules и кликаем в пункт Install new module.

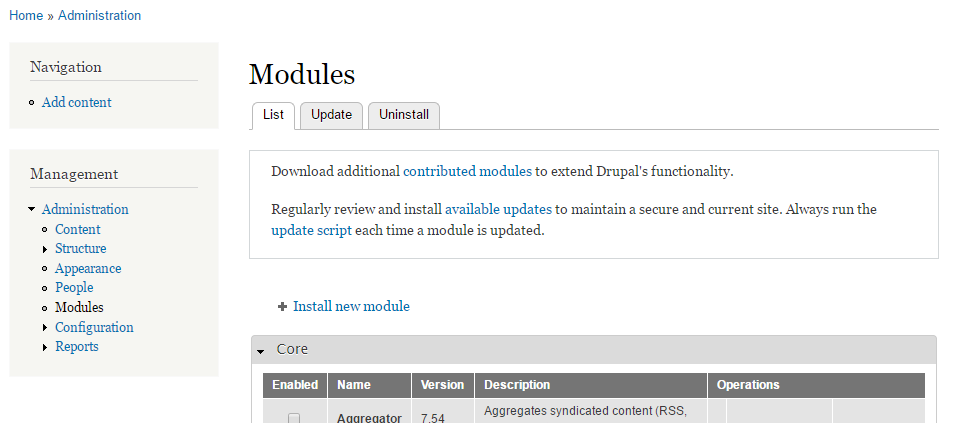


Рисунок – Список модулей

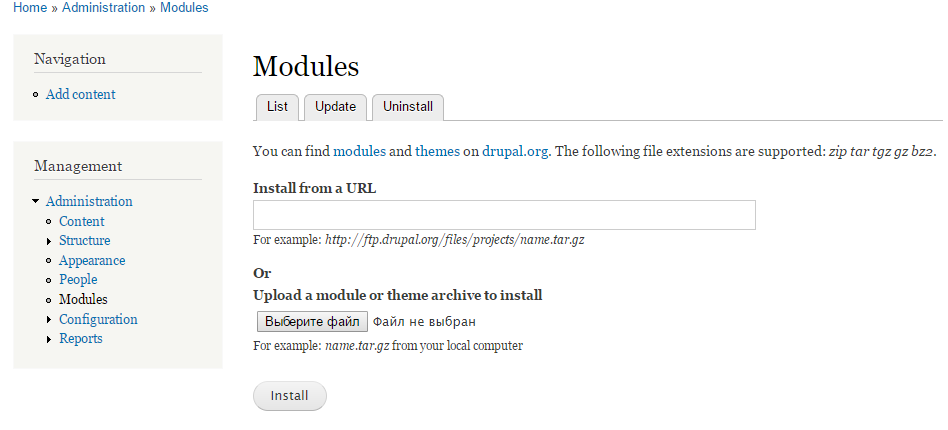


Рисунок – Окно установки модуля

На сайте Drupal.org, в поле имени модуля вводим название модуля из списка выше, потом выбираем нужный в результатах поиска.

На странице модуля можно прочитать его описание, посмотреть документацию, скриншоты и примеры использования.

В самом низу страницы находится список для скачивания модуля для разных версий Drupal. Ссылки также помечены разными цветами. Зеленым цветом выделены ссылки на модули, проверенные официальным разработчиком. Кликаем правой кнопкой мыши (далее ПКМ) по нужной ссылке и выбираем копировать адрес ссылки.

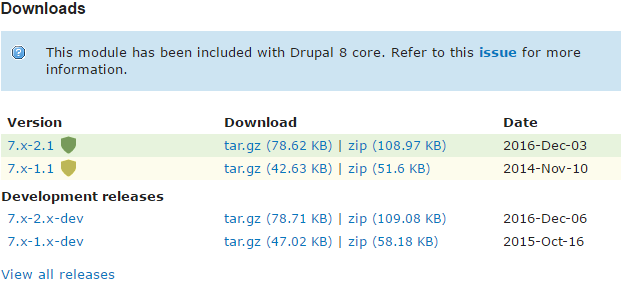


Рисунок – Список ссылок для скачивания

Далее вставляем ссылку, которую мы скопировали ранее в поле Install from url и кликаем по кнопке Install. Далее установка всех модулей, за исключением самописных, будет происходить таким же образом.

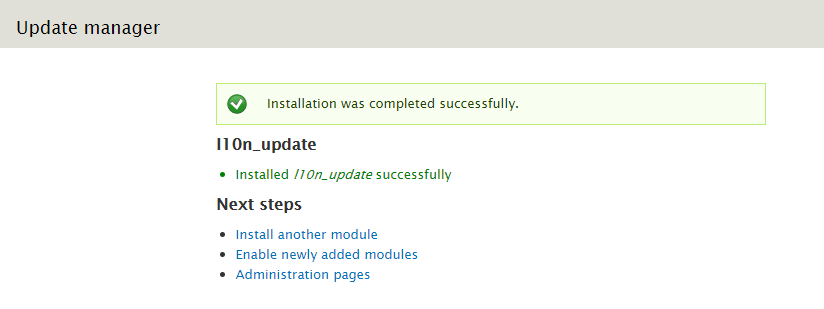


Рисунок – Успешная установка модуля

Теперь, установленный, модуль нужно включить, кликнув Enable newly added modules, выбрать его в списке и сохранить конфигурацию.

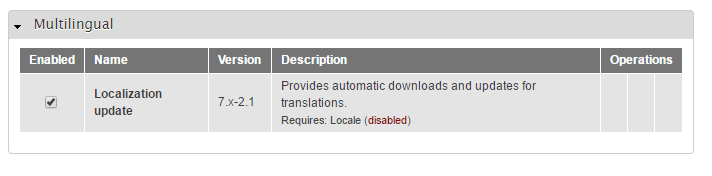


Рисунок – включение модуля

Теперь можно добавить русский язык, переходим во вкладку configuration, далее languages и выбираем пункт add language

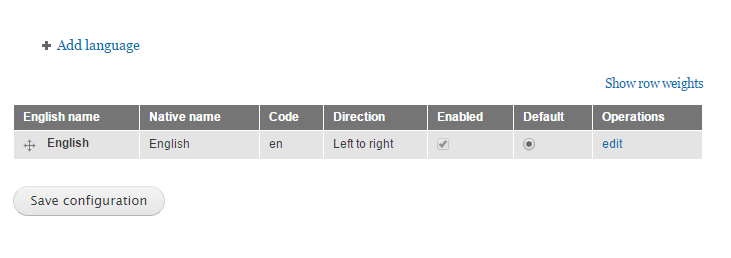


Рисунок – Окно конфигурации языка

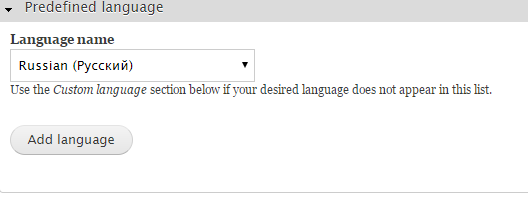


Рисунок – Выбор русского языка из списка

После установки выбираем русский язык по умолчанию и сохраняем конфигурацию

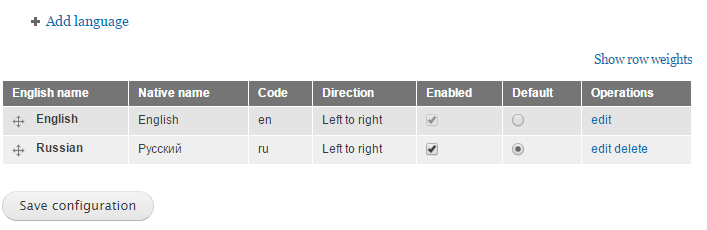


Рисунок – Смена языка по умолчанию

После установки русского языка, все модули, которые уже установлены и будет установлены, будут локализованы, если имеется поддержка данного языка. Далее устанавливаются остальные модули из списка.

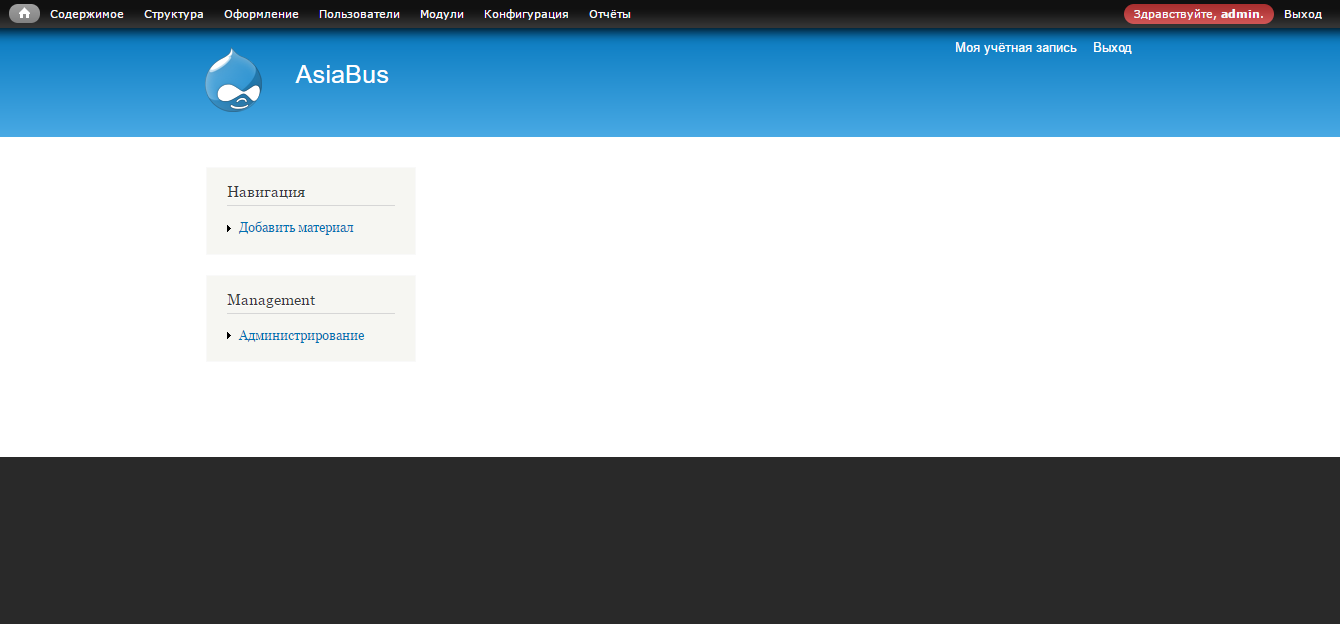


Рисунок – Главная страница после установки модулей.

Теперь есть удобная панель администратора сверху, благодаря которой можно быстро получить доступ к любой функции CMS Drupal. Далее можно приступать к темизации.

3.3.3 Темизация

Темы – это плагины Drupal, которые придают сайту Drupal новый облик.

Тема ­ важный компонент пользовательского интерфейса (UI) Web-сайта на основе Drupal. В версии Drupal 7 появилась новая методология реализации тем, хотя структура темы существенно не изменилась.

Тема Drupal отделяет логику обработки системы от ее элементов дизайна. Для этого Drupal использует сложную структуру тем, которая состоит из тем, механизмов тем и связей. Компоненты темы взаимодействуют с системами ядра Drupal и элементами дизайна модулей, образуя оригинальный пользовательский интерфейс ― отдельные страницы и формы Drupal. Когда бизнес-логика Drupal отделена от логики представления, код легче поддерживать, а реализацию темы можно менять без необходимости переписывать код, и наоборот. Архитектура среды Drupal показана на рисунке.

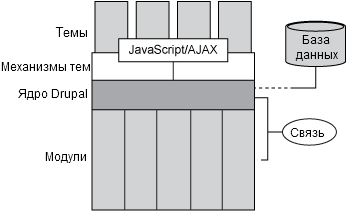


Рисунок – Архитектура среды Drupal

Темы отображают данные, которые ядро Drupal извлекает из базы данных через механизм тем, служащий интерфейсом между ядром Drupal и шаблонами тем.

**Механизм тем**

Механизмы тем ­ это средства взаимодействия тем с ядром Drupal. Существует несколько механизмов тем, которые можно использовать с Drupal. Вот наиболее популярные из них:

* PHPTemplate
* XTemplate
* Plain PHP
* Smarty Engine

При желании можно использовать свой собственный механизм.

Механизм тем Drupal по умолчанию PHPTemplate, который рассматривается в этой статье, использует для функций тем Drupal типа theme\_filename() отдельные файлы тем с такими именами, как filename.tpl.php. Каждый из этих файлов содержит HTML-основу и PHP-операторы для работы с динамическими данными. Таким образом, при наличии начальных знаний в области PHP можно легко создавать достаточно сложные темы с помощью PHPTemplate, так как они содержат всего лишь небольшие фрагменты кода.

**Планирование темы**

Тема состоит из нескольких файлов. Их количество зависит от сложности темы. В теме может быть всего три файла или несколько десятков, включая графические элементы и различные сценарии, размещенные в нескольких папках в каталоге theme. На рисунке 2 показаны типичные описания файлов (цвета, изображения, логотип, шаблоны, таблицы стилей и PHP-файлы), которые могут содержаться в главной папке темы.

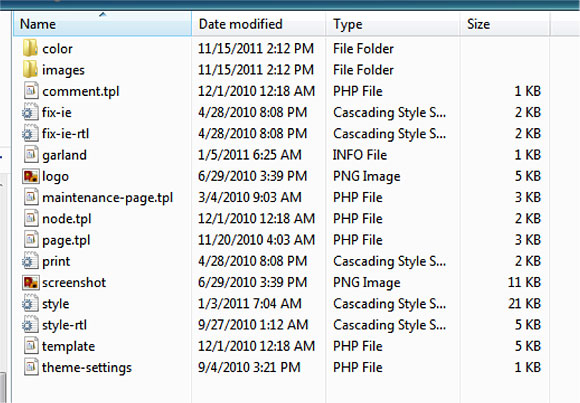


Рисунок ­ Типичный состав папки темы

**Типичный набор папок и файлов в папке темы**

Не все эти файлы необходимы для темы, и в некоторых темах нет даже файла каскадных таблиц стилей (CSS), который многие считают абсолютно необходимым. Структуру темы и то, как она работает, легче понять, когда знаешь, как используются все эти файлы. Рассмотрим некоторые файлы из каталога theme и их функции в рамках общей темы.

**Файл .info**

.info — это обязательный файл: он должен присутствовать в Drupal, чтобы тема была видна. Файлы .info указывают Drupal внутреннее имя темы. например, если имя файла — ibmtheme.info,, то Drupal присвоит этой теме имя ibmtheme. Если в теме используются такие элементы, как JavaScript, метаданные, таблицы стилей или прямоугольные области, их также необходимо определить в файле .info.

**Файлы шаблонов .tpl.php**

В каталоге theme есть несколько файлов шаблонов с именами типа xxx.tpl.php. Эти файлы содержат разметку Extensible HTML (XHTML) и PHP-переменные темы. В некоторых случаях они могут содержать и другие виды выходных данных, такие как RSS. В общем случае каждый файл темы Drupal .tpl.php управляет определенными выходными данными. Встраивать сложную логику в файлы шаблонов неудобно и нерационально, так как это затрудняет их обслуживание. Желательно, чтобы они содержали только прямые XHTML-теги и переменные PHP.

**Файл template.php**

Файл template.php обычно содержит всю логику условных переходов и обработки выходных данных темы. Файл template.php можно использовать также для упрощения файлов .tpl.php темы. Так как это PHP-файл, он обязательно должен начинаться с открывающего тега PHP <?php, но закрывающий тег можно опустить.

**Создание темы**

Существует два основных метода создания темы: можно построить тему с нуля, или изменить уже существующую. Мы создадим новую тему с нуля. Тем не менее, мы будем держать под рукой существующие темы, чтобы видеть, как выглядит их структура.

**Создание структуры каталогов**

Сначала создадим каталог для хранения файлов темы. Самое подходящее место для этого каталога - папка sites/all/themes. Присвоим ему уникальное имя, характерное для нашей темы: оно не должно содержать пробелов.

Хотя это не обязательно, полезно также создать подкаталоги для таблицы стилей (CSS-файлов), изображений и сценариев (если они используются). Это поможет сохранить наглядность.

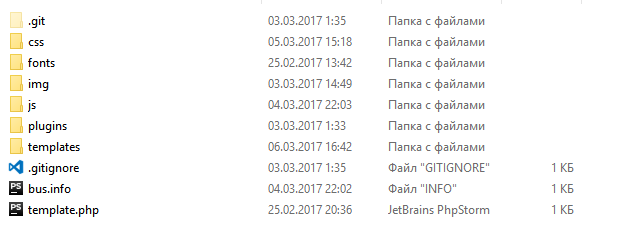


Рисунок – Структура папки с новой темой

В папке css будут храниться файлы стилей, fonts папка для шрифтов, img папка для изображений, js папка для скриптов, plugins папка для сторонних расширений и фреймворков, templates папка для шаблонов страниц и представлений.

**Создание файла .info**

Файл .info ― это просто текстовый файл, содержащий данные ― как правило, параметры, необходимые для описания структуры, а также содержание и параметры настройки темы. Это текстовый файл, каждая строка которого содержит пару из ключа и значения, причем значение расположено справа, а ключ ― слева от знака равенства (=).

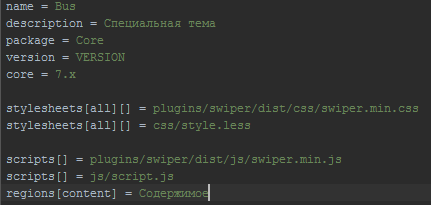


Рисунок – Файл с информацией о теме.

Name – имя темы

Description – описание темы

stylesheet – массив таблиц стилей

scripts – массив подключаемых скриптов

regions – область

После того, как создан файл с расширением .info, в админ-панели можно включить нашу тему и поставить ее по умолчанию.

В верхнем меню кликаем по пункту – оформление и в списке тем выбираем – bus, включаем ее и устанавливаем по умолчанию.

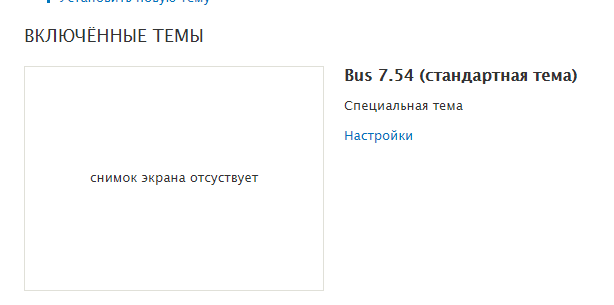


Рисунок – Тема по умолчанию.

Важно для оформления административных страниц выбрать одну из базовых тем, в данном случае это будет тема seven, для того чтобы избежать ошибок и конфликтов при создании материалов и т. д.

**Области**

Ключ regions определяет области блока, доступные для темы. Необходимо определить ключ regions, а затем указать внутреннее машиночитаемое имя в квадратных скобках. После этого нужно в качестве применимого значения указать имя, понятное для человека. Синтаксис:

regions[highlighted] =Mission Statement

В нашем случаем всего одна область regions[content] =Содержимое.

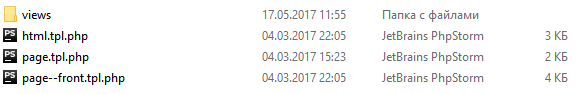


Рисунок – Структура папки tamplates

В данной папке будут храниться все шаблоны документа, внутренних страниц, главной страницы, представлений, форм и т. д.

Файл html.tpl.php в данной теме является корневым шаблоном, в нем указывается тип документа, подключаются стили, скрипты и указывается переменная, в которую будут выводится другие дочерние шаблоны.

Файл page.tpl.php это шаблон для любой внутренней страницы нашего сайта. Здесь располагается вся разметка сайта, которая находится внутри тега body, и указывается регион для содержимого страницы. Вместе с файлом html.tpl.php образуется, так называемый, layout, то есть общая разметка для всех страниц.

page--front.tpl.php шаблон для главной страницы сайта. Так как помимо основного содержимого, в отличии от внутренних страниц сайта, на главной есть другие элементы, (слайдер, блок экскурсий и т. д.), то и сама страница должна отличаться, а потому ей задается более уникальное имя. После сброса кеша на сайте, главная страница будет попадать под более уникальный шаблон (front). Для того чтобы видеть из какого шаблона выводится страница, необходимо изменить настройки сайта. В папке sites/all/themes/default есть файл settings.php, в котором указаны доступы к базе данных, строка подключения и закомментированы некоторые полезные строки.

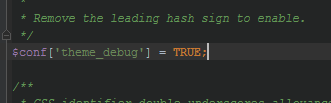


Рисунок – файл settings.php

Убрав комментарий с данной строки, мы сможем увидеть дополнительную информацию о шаблонах, пути к ним и т.д. прямо в браузере, в инструментах разработчика. Крестиком будет указан текущий шаблон.

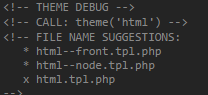


Рисунок – информация о шаблоне

В шаблоне html.tpl.php нет необходимости что-то менять, за исключением типа документа и добавления некоторых meta-тегов.

Далее создается шаблон для главной страницы (page--front.tpl.php) и для внутренних страниц. В каждый из этих шаблонов вставляется верстка, которая повторяется на всех страницах сайта, то есть: хедер (шапка) сайта и футер (подвал). Весь контент, который находится между этими блоками, будет выводится в других шаблонах (нодах, представлениях и т.д.).

После сброса кеша на сайте, мы увидим разметку главной страницы, но без применения стилей, так как drupal, пока, не понимает less. Указав специальную строку в шаблоне (<?php print $messages; ?>), в содержимом сайта будет выводится информация об ошибках, очистке кеша на сайте и т.д. Загрузив страницу без поддержки препроцессора less, мы увидим сообщение о том, что drupal не может загрузить стили и указывает, что нужно для исправления ошибки. Далее устанавливаем модуль LESS CSS Preprocessor, а также, в папку /sites/all/libraries добавляем библиотеку less.php, ссылка на скачивание которой, находится на странице с модулем.

Подключив поддержку стилей, написанных на css-препроцессоре less, в консоли инструментов разработчика браузера Google Chrome, выводятся ошибки, говорящие, что браузер не может найти изображения (логотип в шапке и подвале), так как неверно указан путь. При верстке был указан статичный путь к изображениям, теперь нужно указывать путь относительно корневого каталога сайта (/sites/all/themes/[папка с изображениями]). Для того, чтобы не вводить корневой путь каждый раз, его можно присвоить, как значение переменной, которую так нет необходимости прописывать в ручной. Для этого нам поможет хук preprocess\_page, код которого добавляется в файл template.php.

Как и любой веб-фреймворк, Drupal берет на себя часть обработки HTTP-запроса, вызывая написанные в модулях функции на определенных этапах. Drupal знает, когда нужно вызвать ту или иную функцию некоторого модуля благодаря механизму хуков(hooks). Хук – это спецификация API-функции Drupal с фиксированным именем, сигнатурой и семантикой. Реализация хука – это PHP-функция в некотором модуле Drupal, соответствующая спецификации.По смыслу хук аналогичен интерфейсу в ООП, а реализация хука – реализации интерфейса в некотором классе. Однако возможности ООП PHP не используются для осуществления работы хуков.

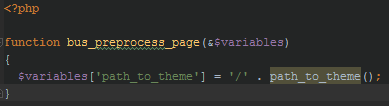


Рисунок – содержимое файла template.php

На рисунке показан хук bus\_preprocess\_page, который срабатывает при загрузке страницы. В массив $variables добавляется элемент path\_to\_theme, и присваивается значение пути до темы, при помощи функции path\_to\_theme().

Теперь в шаблонах внутренней страницы и главной страницы пути до изображений можно указать следующим образом:

<img src=”<?php print $path\_to\_theme; ?>/img/[имя].расширение” alt=””/>

После проделанных выше операций, на странице сайта будут применяться стили и корректно выводится изображения, но на странице все еще нет контента.

3.3.3.1 Создание типов материалов

В drupal все (каждая страница, блок, элемент блока, форма и т.д.) является сущностью/объектом. Соответственно, у каждой сущности есть свои атрибуты, которые отличают ее от других. В нашем случае примером сущности является тур, атрибутами которого будут: тип сущности (тур), изображения, название, описание, страна. Для того, чтобы добавить содержимое на сайт (туры, экскурсии, страны и т.д.) в drupal есть возможность создания своих сущностей или типов материалов.

Для примера создадим тип материала автобус:

1. Переходим в пункт меню структура -> типы материалов -> добавить тип материала

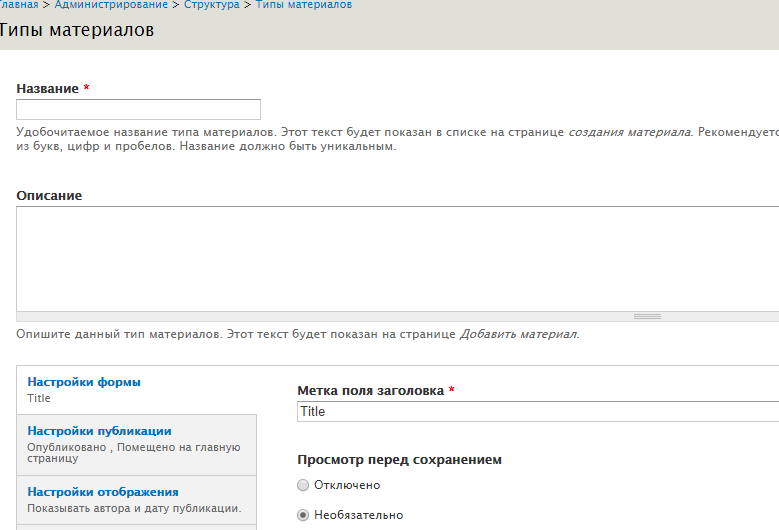


Рисунок – добавление типа материала.

В поле название вводим логичное имя для нашего типа материала (Автобус) и машинное имя bus. Поле описание можно проигнорировать и оставить пустым, метку заголовка лучше исправить на «Заголовок», так как администратору сайта в будущем, проще понимать русский язык. Во вкладке настройка публикации отключаем публикацию на главной странице, в настройке отображения убираем вывод даты и автора публикации, в настройке меню можно убрать добавление пункта в меню, так как для данного типа материала в этом нет необходимости. Сохраняем тип материала и добавляем поля.

1. Добавление полей

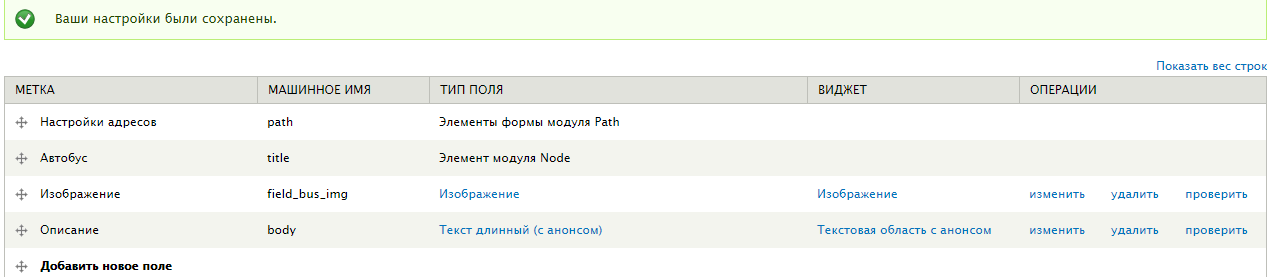


Рисунок – Страница добавления полей

По умолчанию у типа материала уже существуют поля title, body и поле path, которое появляется при включении модуля path. Модуль path является модулем ядра drupal, устанавливать его не нужно. По psd-макету внутренней страницы автобуса можно выделить следующие поля:

* Изображение;
* заголовок;
* описание.

Поля стоимость, минимальное время аренды и количество мест, по желанию заказчика, можно опустить.

В строке добавить новое поле вводим имя для нашего поля (Изображения), машинное имя (field\_bus\_img) (желательно указаывать машинное имя, именно в таком формате field\_[машинное имя материала]\_[имя поля], это поможет избежать дальнейшего конфликта между материалами, так как поле изображений может быть и у других сущностей). В настройках поля выбираем его обязательным, директорию для загрузки изображений также необходимо указать, чтобы структурировать изображения по типам материалов на сервере, и выбираем неограниченное количество для поля, чтобы можно было загрузить несколько фотографий.

Остальные поля уже существуют, можно сохранить тип материала и добавить несколько автобусов.

Переходим в пункт меню Содержимое -> добавить материал -> Автобус

Вводим названия для нового автобуса, изображение для него. В поле описание вводим информацию об автобусе. Но что, если нам необходимо создать список, выделить текст жирным или курсивом, ведь в drupal изначально нет текстового редактора, а обычный пользователь не всегда может знать html-теги. Для этого понадобится модуль CKeditor, который скачивается отдельно. После его установки и включению в пункте меню Конфигурация -> Работа с содержимым -> Текстовые форматы добавляем новый формат, называть его принято FULL text

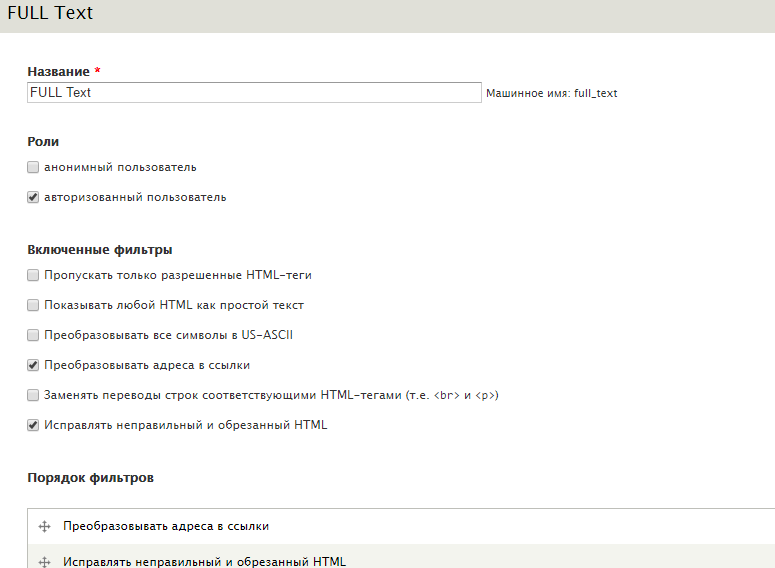


Рисунок – Настройка нового текстового формата

Настраиваем CKEditor (Конфигурация -> Работа с содержимым -> CKEditor), добавляем новый профиль и выбираем для него текстовый формат.

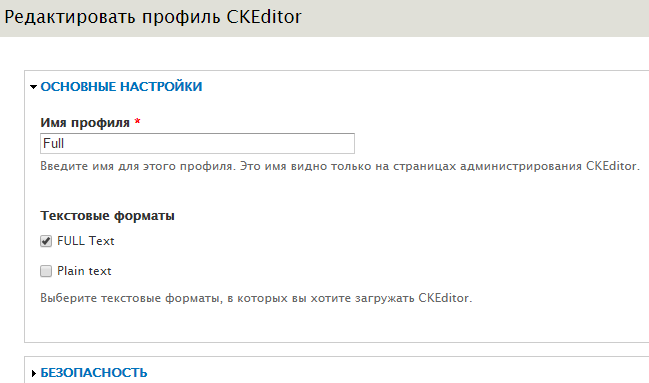


Рисунок – Создание профиля в CKEditor

Теперь при создании материала, в полях с типом – Длинный текст, появился WYSIWYG-редактор.

WYSIWYG (произносится [ˈwɪziwɪɡ], является аббревиатурой от англ. What You See Is What You Get, «что видишь, то и получишь») — свойство прикладных программ или веб-интерфейсов, в которых содержание отображается в процессе редактирования и выглядит максимально близко похожим на конечную продукцию, которая может быть печатным документом, веб-страницей или презентацией. В настоящее время для подобных программ также широко используется понятие «визуальный редактор».

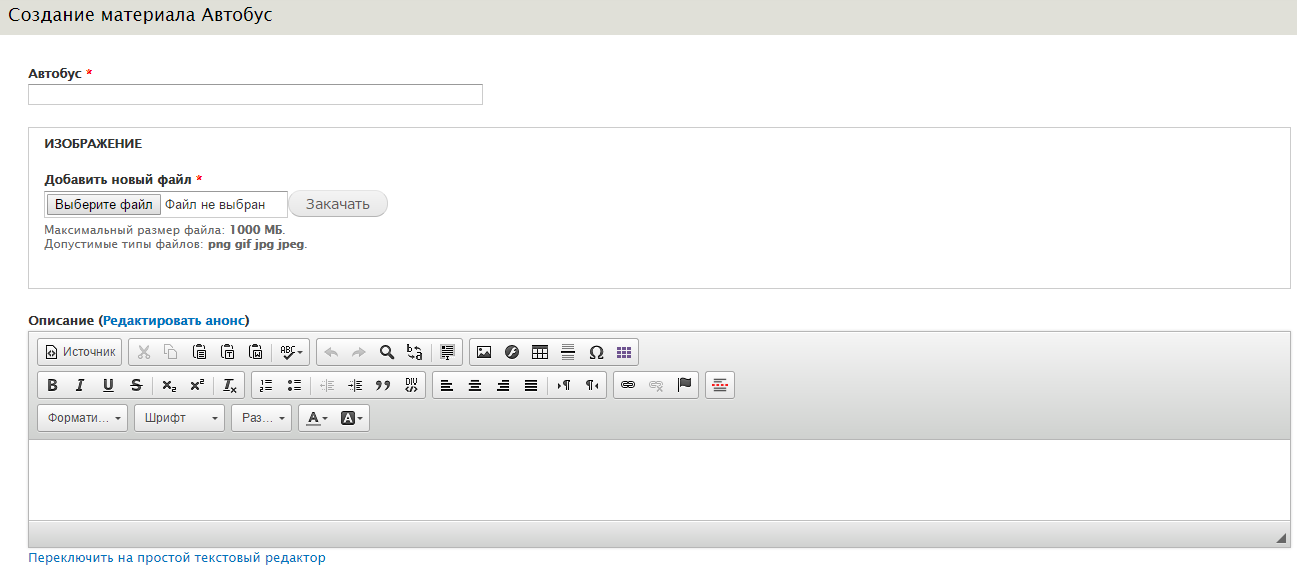


Рисунок – Окно создания материала

После создания материала, мы автоматически попадаем на страницу с его полным содержимым. Шаблоном является node.tpl.php из ядра drupal.

Далее все типы материалов будут создаваться по такому же сценарию, отличаться будут только их названия и поля.

3.3.3.2