[ВВЕДЕНИЕ 3](#_Toc485727824)

[1. Общие сведения ООО «Перфоманс» 3](#_Toc485727825)

[1.1 Основные виды деятельности 3](#_Toc485727826)

[1.2 SWOT-анализ 10](#_Toc485727827)

[2 Разработка сайта для туроператора Азия Бас 3](#_Toc485727828)

[3 Создание темы для CMS Drupal 7 3](#_Toc485727829)

[3.3 Темизация 12](#_Toc485727830)

[3.4 Создание типов материалов 21](#_Toc485727831)

[3.5 Создание представлений 25](#_Toc485727832)

[ЗАКЛЮЧЕНИЕ 3](#_Toc485727833)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 3](#_Toc485727834)

ВВЕДЕНИЕ

1. Общие сведения ООО «Перфоманс»

Перфоманс – крупнейшая веб-студия Хабаровского края. Существует с 2007 года и занимается созданием, продвижением сайтов, дизайном, интернет-рекламой.

1.1 Основные виды деятельности

**Разработка сайтов**

Создание, развитие, продвижение – три простых слова описывающие весь цикл веб-компаний, занимающихся разработкой и созданием сайтов. Каждое из этих слов включает множество действий и дел. В зависимости от текущего состояния вашего веб-сайта этапы могут меняться местами, но начинается все с создания сайта.

В Хабаровске достаточно веб-студий, которые предлагают разработку сайтов. Но Перфоманс предлагает именно качественное создание сайтов.

Стандарты качества, которых придерживается веб-студия «Перфоманс»:

* В основе – достоверная информация

Прежде чем приступить к работе, разработчику необходимо знать о целях, которые преследует заказчик, создавая сайт. При этом источником информации является не только сам заказчик, но и вся область отрасли в целом.

Чем больше качественных данных получится собрать, тем более качественным будет создание сайта. Информация нужно собрать, зафиксировать, запротоколировать и проанализировать. В итоги рождается качественное техническое задание, где освещены не только, к примеру, программная платформа и карта сайта, а абсолютно все аспекты и бизнес-задачи будущего веб-сайта от подробного описания портрета посетителя сайта до перечня браузеров, в которых должен отображаться сайт.

* Организованность

У каждого производства, будь то выпечка булочек или проведение предвыборной компании, должен быть руководитель, который и отвечает в конечном итоге за качество созданной продукции. Так же и у каждого проекта обязательно должен быть руководитель проекта со стороны веб-разработчика или проект-менеджер. Руководитель веб-проекта ­ это отдельный высококомпетентный сотрудник, который не программирует сайт, не рисует его, а настраивает и обеспечивает взаимодействие всех служб и представителей заказчика сайта с командой разработчика. При этом он генерирует идеи по разработке сайта и одновременно занимается их реализацией. Наши проект-менеджеры всегда в курсе всего, что касается данного проекта.

* Каждый сайт уникален

Уникален каждый человек, заказывающий создание сайтов. Хабаровск ­ деловой город, здесь много компаний, но каждая из них имеет свою индивидуальность. Мы умеем углядеть эту особенность компании, даже если она не бросается в глаза и не видна самому заказчику. Ведь эксклюзивность — это конкурентоспособность прежде всего. Для этого нужно быть искренне заинтересованным в разработке и создании неповторимого сайта, иметь пытливый ум, не лениться во все вникать и иметь нестандартное мышление. Мы относимся к каждому веб-сайту, как к замечательной возможности реализовать что-то новое и отличное от других.

* Дизайн: говорящие идеи

Всё, что существует в техническом задании, в мыслях заказчика и разработчика, визуализируется в конкретном концепт-макете сайта. Дизайн сайта должен учитывать все, что было увидено и услышано ранее. Дизайнер может создать что-то типовое или что-то эксклюзивное — главное, чтобы осуществлялись все поставленные задачи. Опять же, в своей работе дизайнер ориентируется не на работу Хабаровских веб-студий, а как минимум на весь Рунет. Дизайн говорит на международном языке, встречает посетителя, информирует его, направляет именно туда, куда гость сайта сам желает попасть.

* Программирование — арсенал решений

Для окончательной программной реализации качественного сайта мы имеем широкий арсенал технических решений. Мы работаем с самыми популярными системами управления контентом. У нас огромный архив готовых и проверенных временем и пользователями решений. Но даже если нет готового решения, мы сумеем быстро сориентироваться и найти его, ведь в распоряжении всегда мировая коллекция накопленных знаний в сети интернет.

Продвижение сайтов

Современный мир не стоит на месте. Хабаровск – большой город, развивающийся стремительными темпами. Для успешного продвижения на рынке Хабаровска, каждой компании необходимо донести до потребителей информацию о себе. Сегодня самым популярным и доступным СМИ является интернет. С каждым днем количество интернет-пользователей растет, а интернет становится быстрее, доступнее и дешевле. Поэтому создание и последующее продвижение сайта в интернете – это отличная возможность для развития бизнеса.

Раскрутка сайтов и поисковая оптимизация сайтов в Хабаровске ­ визитная карточка большей половины web-студий нашего города.

5 причин, почему следует обратиться в компанию «Перфоманс» для раскрутки сайта:

1. Семилетний опыт успешной работы.

Среди всех компаний всегда есть новички - те, кто только вышли на рынок и только начинают поиски эффективных решений и стратегий. Насколько успешными или нет будут эти эксперименты станет известно только спустя 3-4 месяца, и в любом случае это будут эксперименты за ваши деньги.

Мы работаем в сфере IT-услуг уже более 9 лет! За это время к нам обратилось множество клиентов, которые довольны сотрудничеством, среди них и такие известные, как завод Дальэнергомаш, фабрика окон «Отрис» и многие другие. Благодаря нашим усилиям, все больше людей находят сайты этих компаний и приобретают их услуги. Мы гордимся своими достижениями и дорожим своей репутацией. Наши клиенты продлевают договоры на оказываемые услуги из года в год, потому что им выгодно и комфортно сотрудничать с нами.

2. Оплата за результат.

Наши клиенты платят только за результат. Мы не ограничиваемся в объемах работы и не берем дополнительной платы за нее. Мы делаем столько, сколько требуется для получения реального результата. Вы платите не за количество наших действий, а только за конечный результат. А результат это вывод ключевых запросов вашего сайта на топовые позиции поисковых систем Яндекс и Google или, иначе говоря, в ТОП-10. Это значит, что за 11 место вы уже не платите. Не важно, сколько мероприятий мы проведем для поисковой оптимизации и раскрутки сайтов в Хабаровске или даже по всей России - вы заплатите только за конечный результат. Схема оплаты за результат четко прописана в договоре.

3. Прозрачная схема работы.

Наши методы работы прозрачны и понятны. Вы будете знать, чего мы хотим добиться, и какой объем работ проделаем для этого. Мы используем только законные «белые» схемы поискового продвижения сайтов, это гарантирует высокие позиции на протяжении долгого периода. Каждый месяц вы будете получать полный детальный отчет о проделанной работе и ее результатах. В зависимости от типа услуги отчет содержит:

* Количество посетителей сайта и количество просмотров сайта;
* Количество платных кликов и их среднюю стоимость;
* Конверсия сайта и эффективность рекламы;
* Ключевые фразы, по которым посетители приходят на ваш сайт;
* Географию посетителей;
* Путь посетителя по страницам и время нахождения на этих страницах;
* Стоимость оптимизации сайта под каждый ключевой запрос в зависимости от его позиций в поисковых системах.

В заключение отчета мы указываем цели и задачи, которые ставим перед собой в следующем периоде.

Схема оплаты наших услуг проста, понятна и четко прописана в договоре. Мы не выставляем скрытых платежей. А если у вас все же возникнут вопросы, то ваш личный менеджер, оперативно ответит на них по телефону, письменно или лично.

4. Комплексный подход.

Компания «Перфоманс» — веб-студия полного цикла. Перед тем, как приступить к работе, мы проверяем сайт вашей компании и даем рекомендации по его улучшению, если это необходимо. Мы проделываем всю необходимую работу самостоятельно. В Хабаровске мы одна из немногих компаний, которые предлагают комплексный подход. Работа ведется над содержимым сайта, над юзабилити и над дизайном страниц. Все это необходимо для повышения конверсии — превращения посетителей сайта в ваших клиентов. Мы точно знаем, что сделать, чтобы посетитель стал клиентом.

Для раскрутки сайтов мы используем полный спектр инструментов интернет-маркетинга: медийная и контекстная реклама, интеграция с социальными сетями, работа с форумами, интернет-PR. С нами ваш сайт станет уникальным и полезным. Раскрутка сайта неминуемо ведет к увеличению прибыли вашей компании, повышению осведомленности потенциальных потребителей и лояльности клиентов. За увеличением посещаемости сайта неминуемо следует и увеличение прибыли вашей компании. Мы делаем ваших потенциальных потребителей – реальными.

5. Технологии индивидуальности.

Шаблонный подход к проекту не сможет дать эффективного результата. Для каждой компании, для каждого товара необходим индивидуальный подход. Исходя из миссии, целей и задач компании мы разрабатываем индивидуальную стратегию продвижения сайта. Сначала мы полностью изучим ваш бизнес, товар, ваших потребителей и конкурентов, и проведем маркетинговый анализ сайта. И только потом займемся разработкой стратегии раскрутки сайта.

К каждому проекту прикрепляется личный менеджер. Он поможет решить вопросы, связанные с поисковой оптимизацией сайта, продвижением и раскруткой сайта, а также любые технические и маркетинговые вопросы.

Виды продвижения:

* Поисковое (СЕО)

Стоимость — от 10 000 рублей/месяц

зависит от количества словосочетаний, по которым вы хотите, чтобы вас находили, и их конкурентности. Составим для вас бюджет исходя из задач и статистики по вашим конкурентам.

Сроки — 2-3 месяца

появление в топ-10 Яндекса и Гугла в течении 2-3 месяцев. Сроки обусловлены технологическими особенностями.

Это основной вид поискового продвижения, который обеспечит вам максимум клиентов.

* Контекстное (Яндекс.Директ и Google.AdWords)

Стоимость — от 300 рублей за рекламную компанию

зависит от количества словосочетаний, по которым вы хотите, чтобы вас находили и их конкурентности.

Сроки — старт сразу в день пополнения баланса

Снятие денег за фактический переход на ваш сайт.

Вас видно наверху в поиске и на тысячах сайтов, состоящих в рекламной сети.

Удобно для быстрого старта.

Экономим бюджет в 1,5-2 раза. Качественно вести кампанию может только специалист, не смотря на внешнюю простоту интерфейса.

* В социальных сетях

Стоимость

зависит от количества словосочетаний, по которым вы хотите, чтобы вас находили и их конкурентности. Можно настроить списание средств за показ или за переход.

Сроки — старт сразу в день пополнения баланса

Точный выбор аудитории для показа: пол, возраст, регион, интересы и т.д.

* Медийная реклама

Стоимость — от 15 000 рублей/месяц

зависит от площадки, количества показов, тематики. Списывается за показы.

Сроки — ближайший свободный период

обычно старт в течение 1-2 недель.

В основном федеральные площадки: Яндекс.почта и Mail.почта, тематические разделы.

Нужно рисовать и адаптировать баннер.

Это имиджевая реклама, статистика целевых переходов невысокая.

Очень важно правильно подобрать площадки и время показов.

* Специализированные площадки

Стоимость ­ от 3 000 рублей/месяц

зависит от площадки, количества показов, тематики. Списывается за показы.

Сроки ­ ближайший свободный период

обычно старт в течение 1-2 недель.

Целевая площадка дает хороший отклик.

Нужно рисовать и адаптировать баннер.

Очень важно правильно подобрать площадки.

* Любая другая реклама
  + - Подготовим медиаплан по эффективному продвижению вашей компании:
    - массовые рассылки;
    - целевые рассылки;
    - статьи на разных площадках;
    - реклама в тематических роликах на YouTube;

1.2 SWOT-анализ

2 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.

2.1 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.

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.

2.2.1 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

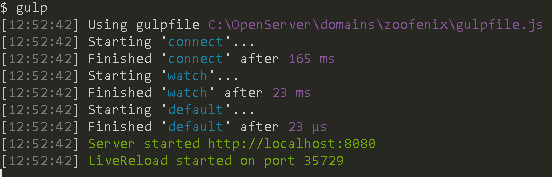


Рисунок – выполнение команды gulp

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

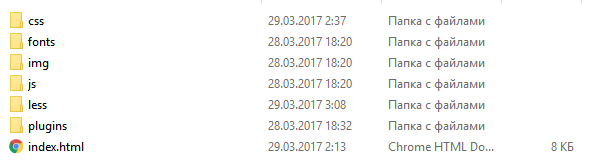


Рисунок – содержимое папки app

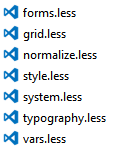


Рисунок – Содержимое папки less

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.

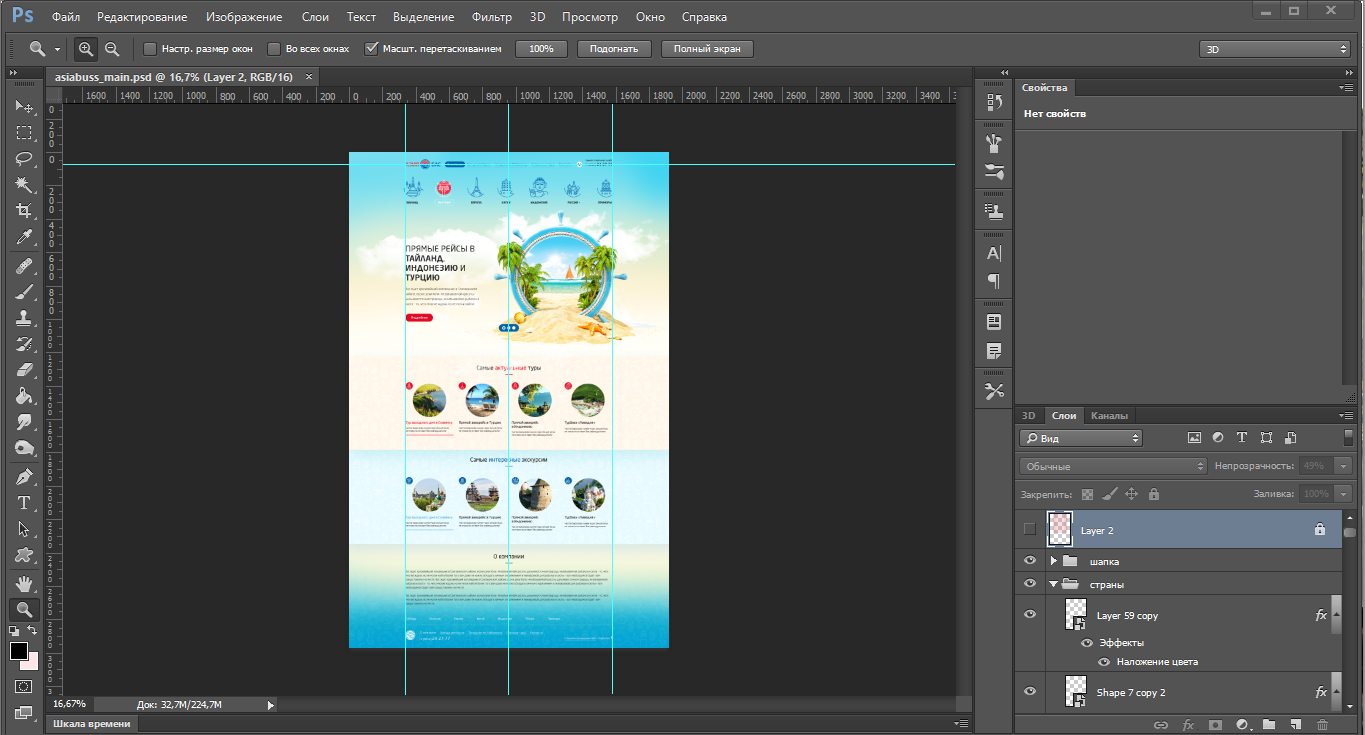


Рисунок – psd-макет в Adobe Photoshop

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

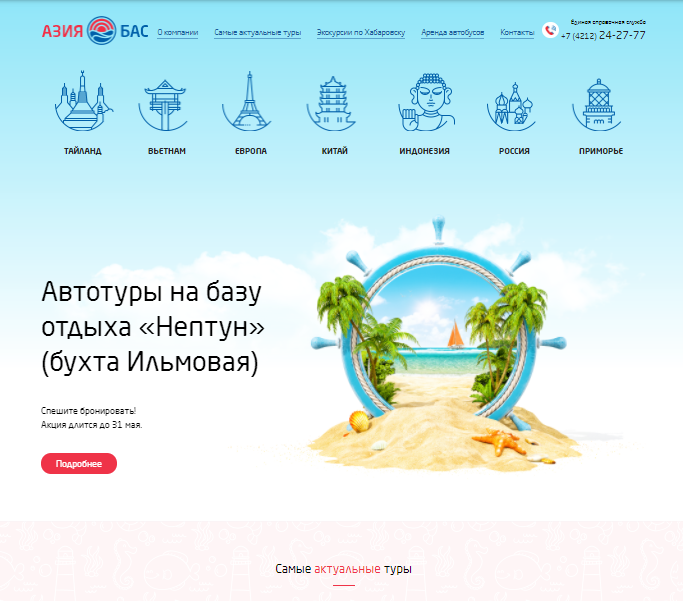


Рисунок – Результат верстки

3 Создание темы для CMS Drupal 7

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.

3.1 Установка Drupal

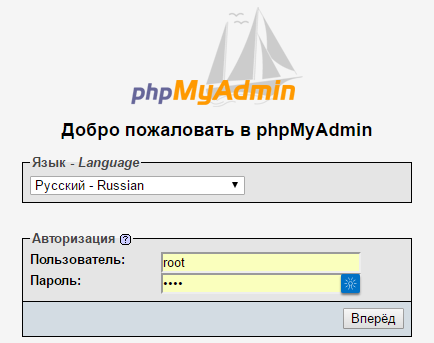


Рисунок – Окно входа в phpMyAdmin

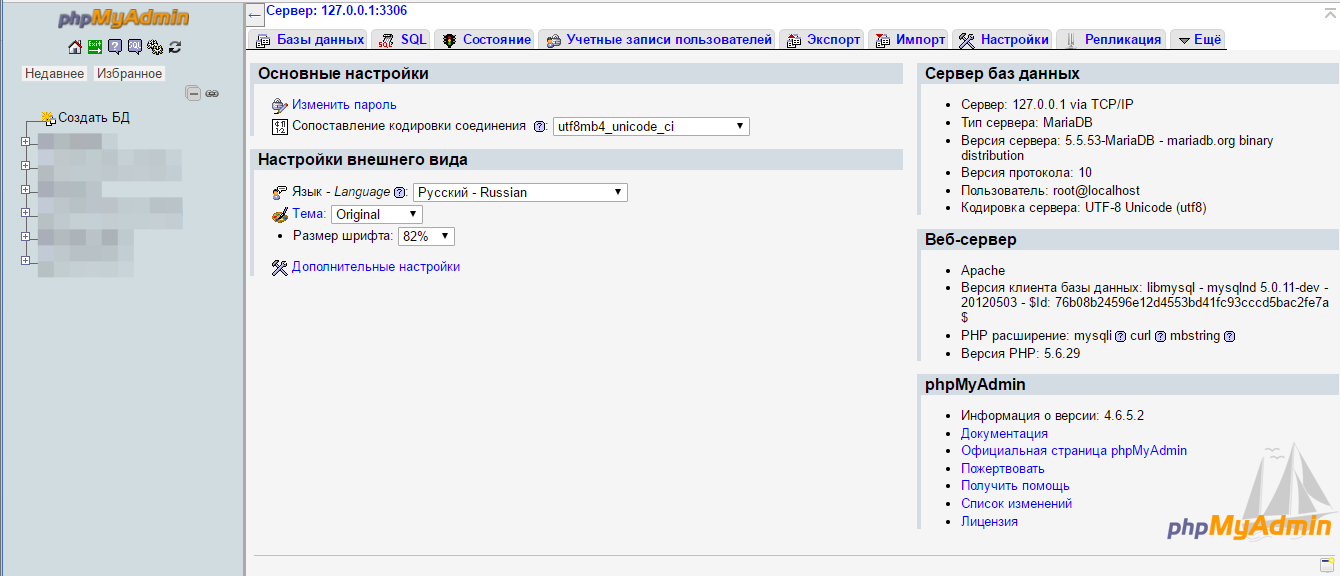


Рисунок – Главное окно phpMyAdmin

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

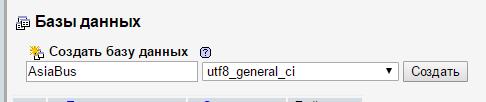


Рисунок – Создание новой БД

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.

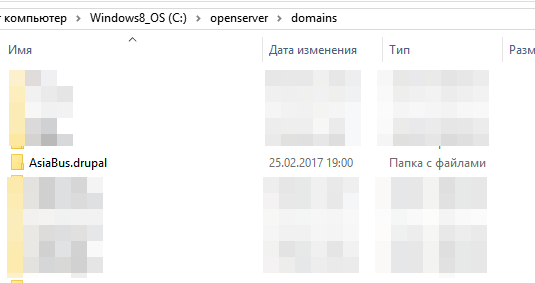


Рисунок – Папка для содержимого сайта

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.

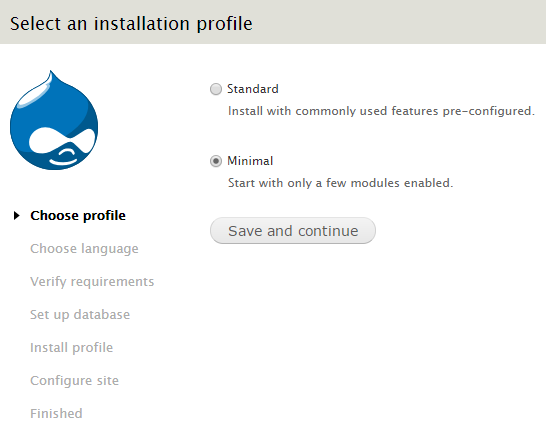


Рисунок – Выбор профиля установки

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

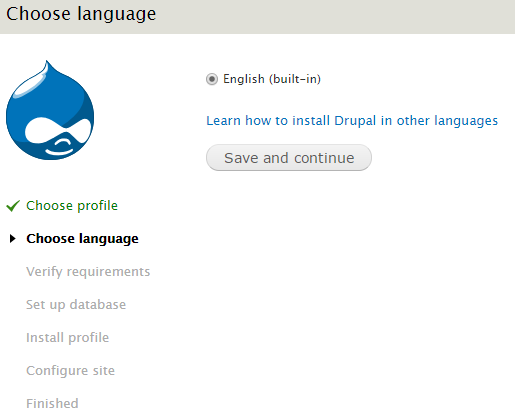


Рисунок – Выбор языка

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

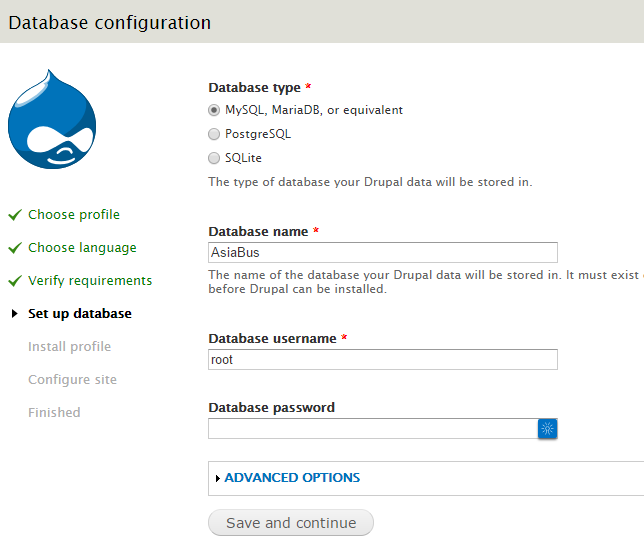


Рисунок – Подключение к базе данных

После идет установка минимального профиля.

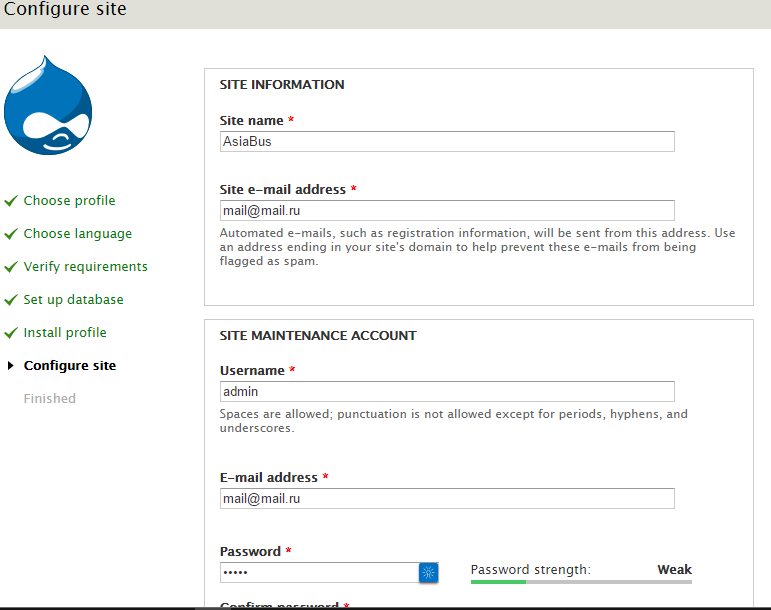


Рисунок – Конфигурация сайта

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

CMS Drupal 7.54 Установлена

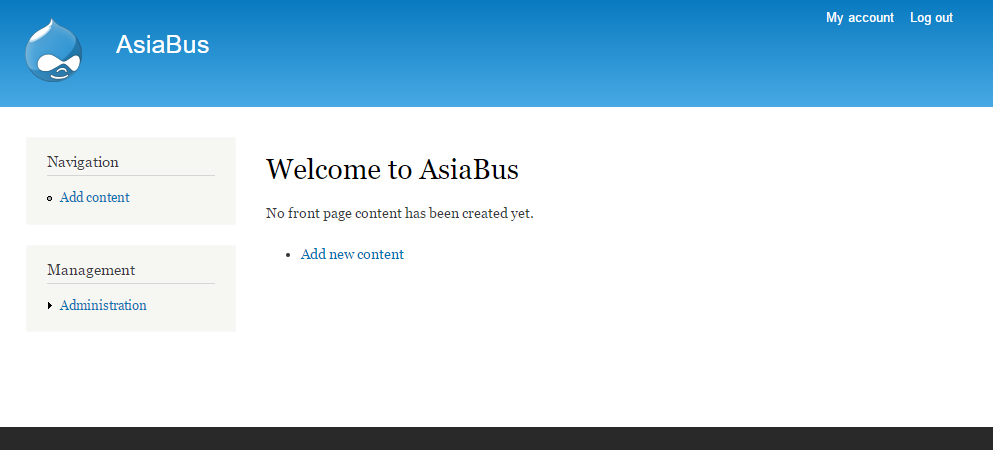


Рисунок – Главная страница сайта

3.2 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.

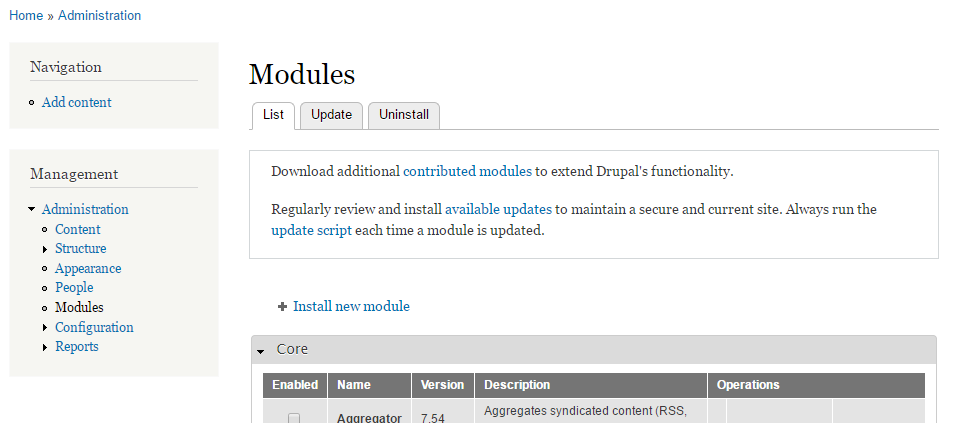


Рисунок – Список модулей

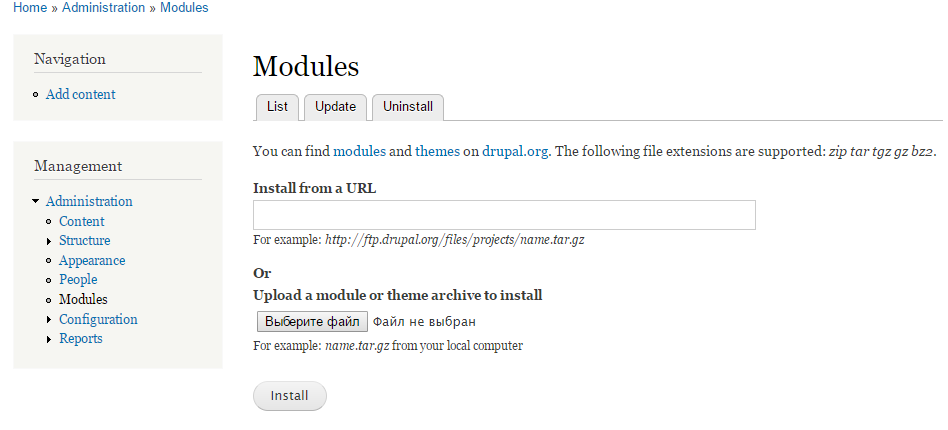


Рисунок – Окно установки модуля

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.

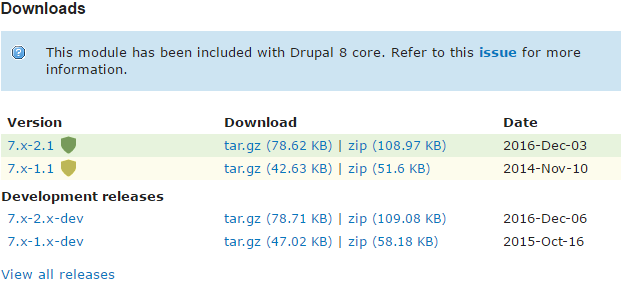


Рисунок – Список ссылок для скачивания

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.

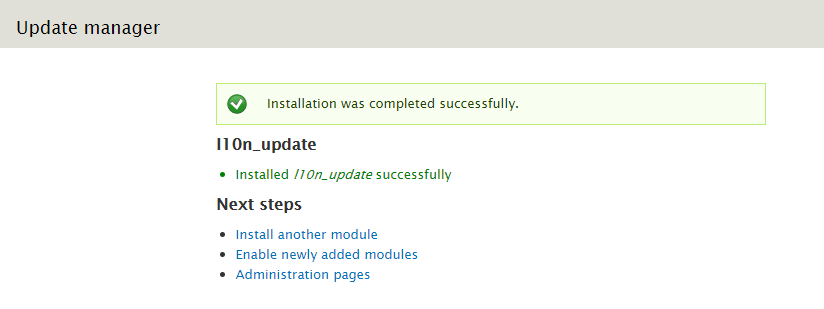


Рисунок – Успешная установка модуля

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

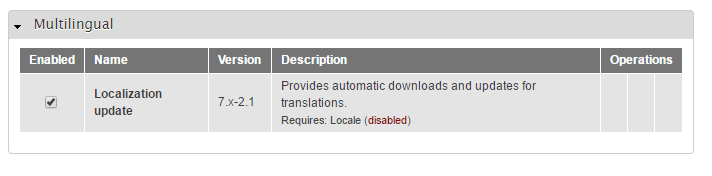


Рисунок – Включение модуля

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

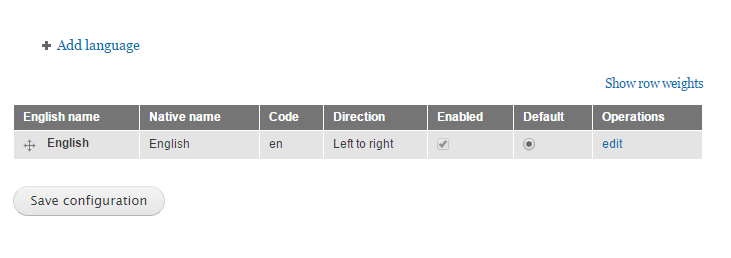


Рисунок – Окно конфигурации языка

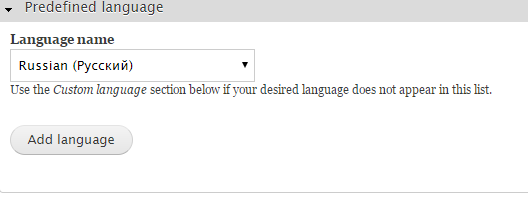


Рисунок – Выбор русского языка из списка

После установки выбираем русский язык по умолчанию и сохраняем конфигурацию

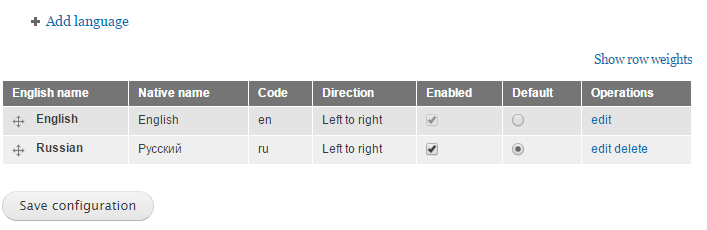


Рисунок – Смена языка по умолчанию

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.

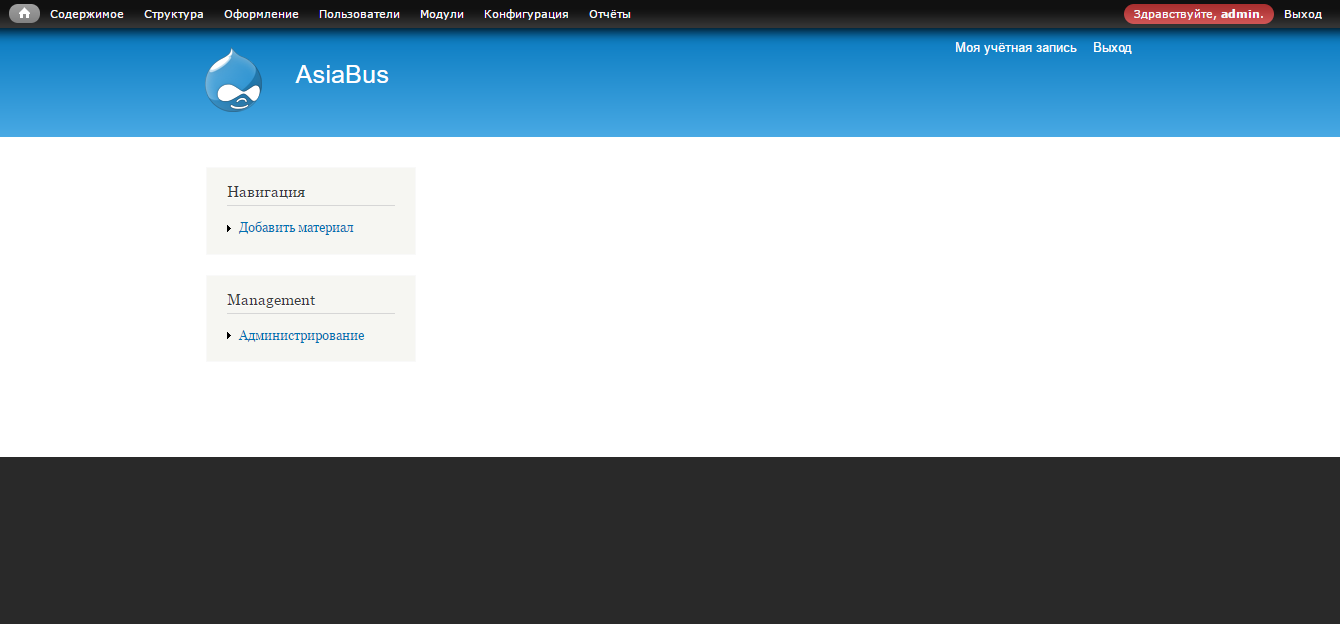


Рисунок – Главная страница после установки модулей

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

3.3 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.

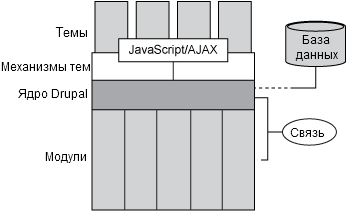


Рисунок – Архитектура среды Drupal

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.

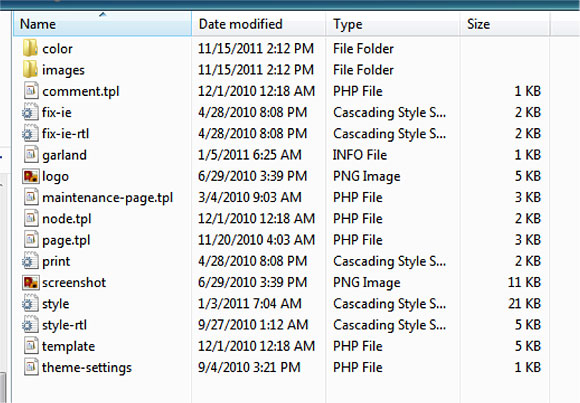


Рисунок ­ Типичный состав папки темы

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

Файл с расширением .info обязателен, в нем находится вся информация о теме, подключаются файлы скриптов и таблицы стилей, выбираются обласи для вывода блоков страниц. Перед расширением находится название темы, создав этот файл, в админ части Drupal, в разделе оформления в списке тем будет новая тема с таким же названием, как info-файл.

Файлы шаблонов .tpl.php

В каталоге темы есть несколько файлов шаблонов с именами типа [имя шаблона].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

В файле template.php содержится вся логика условных переходов и обработки выходных данных темы. Здесь можно добавить переменные для хранения телефонов, электронной почты и т.д. то есть значения, которые могут появляться на нескольких страницах сайта. Так же здесь можно прописывать хуки, особые функции, о которых будет подробнее рассказано далее, при создании шаблона главной страницы.

Создание темы

Тему можно создать несколькими способами:

1. с нуля;
2. на основе существующей.

В данной работе будем применен первый способ, но также можно использовать файлы из существующих тем и изменять их под новую тему (шаблоны).

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.

Новую тему необходимо создавать по следующему пути, /sites/all/themes, следуя от корня сайта. В директории создается новая папка с названием темы. Внутри нее будут лежать все остальные файлы для темы (css, js, изображения, шрифты, шаблоны и т.д.)

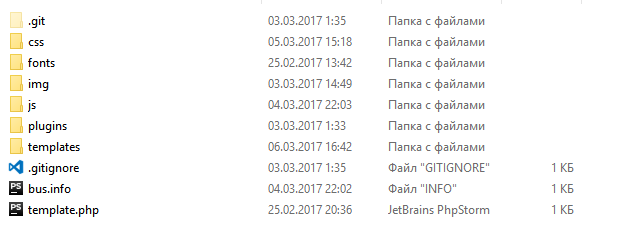


Рисунок – Структура папки с новой темой

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 (=).

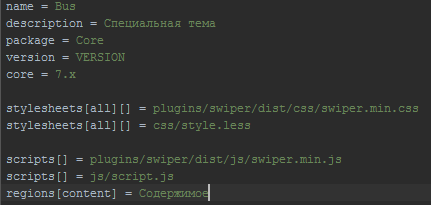


Рисунок – Файл с информацией о теме

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.

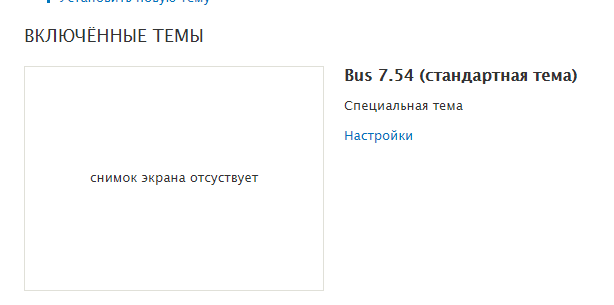


Рисунок – Тема по умолчанию

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[name] = Content

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

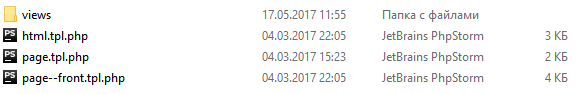


Рисунок – Структура папки tamplates

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.

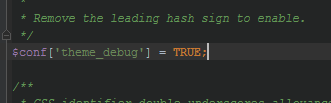


Рисунок – файл settings.php

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.

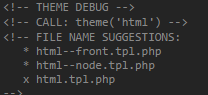


Рисунок – информация о шаблоне

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.

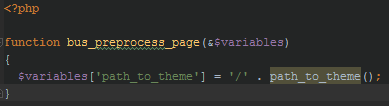


Рисунок – содержимое файла template.php

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.

3.4 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

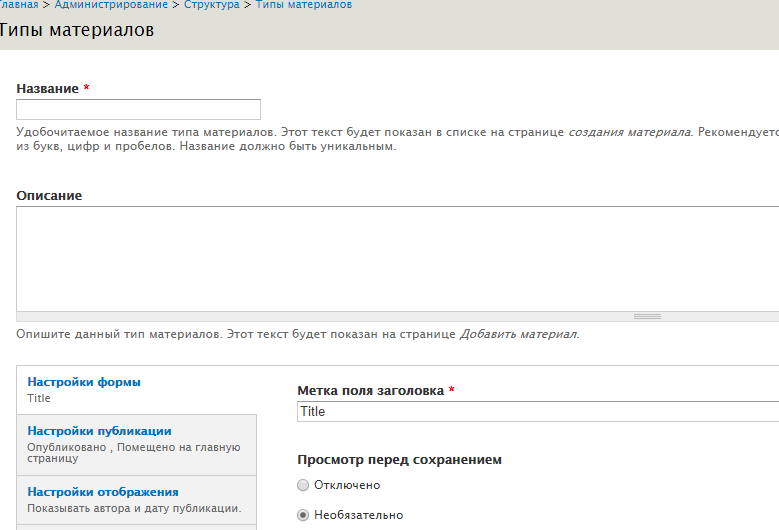


Рисунок – Добавление типа материала

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

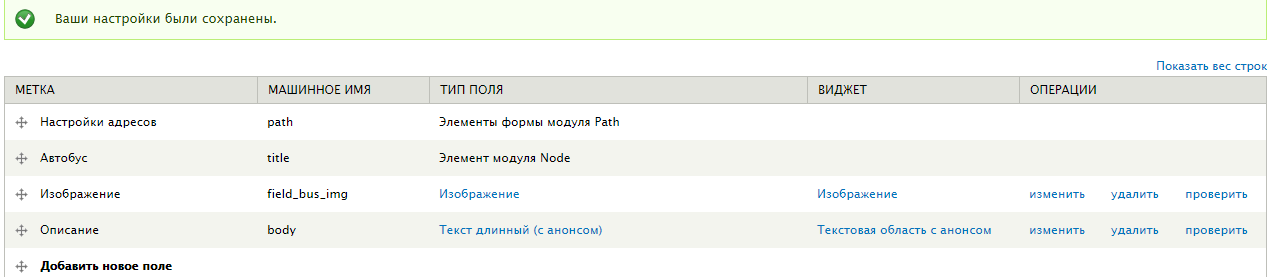


Рисунок – Страница добавления полей

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

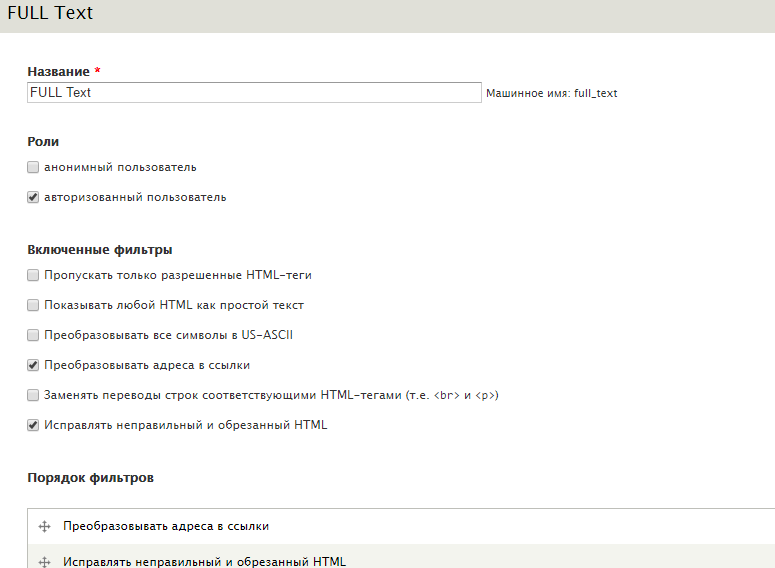


Рисунок – Настройка нового текстового формата

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

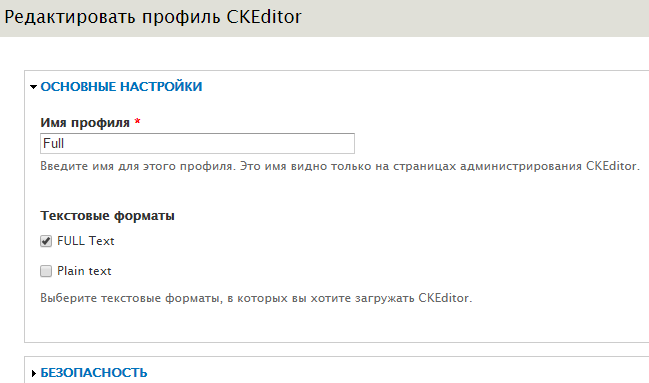


Рисунок – Создание профиля в CKEditor

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».

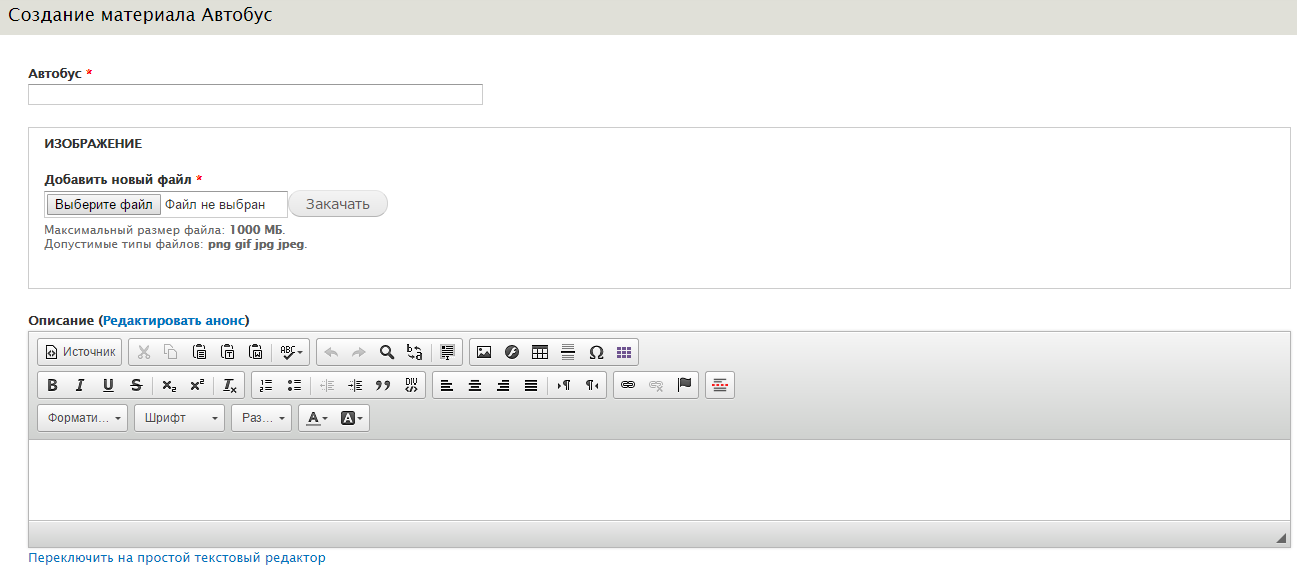


Рисунок – Окно создания материала

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.

3.5 Создание представлений

В этом разделе будет показан один пример для создания представления, при помощи модуля views, самого популярного модуля для cms Drupal, который, как было сказано выше, будет установлен с официального сайта Drupal. Данный модуль позволяет нам выводить на странице несколько материалов одного типа. Как пример можно привести каталог любого интернет-магазина, где однотипными материалами будут являться товары, которых несколько на странице.

Создадим представление для вывода туров. Для этого нужно пройти в пункт меню Структура –> Представления -> Добавить новое представление. В открывшемся окне указываем название нашего представления, его машинное имя, тип материала (ы данном случае Тур), сортировку необязательно, путь к странице (tours). Выбираем создать страницу или блок, либо и блок и страницу. В нашем случае понадобиться только страница. Создание блока будет разобрано далее, на примере создания слайдера. Хотя создание блока ничем не отличается от создания страницы, на примере слайдера будет рассмотрен еще один модуль, добавляющий новый тип полей, при создании материала, и еще одна особенность блока в том, что его можно вывести в любом месте сайта.

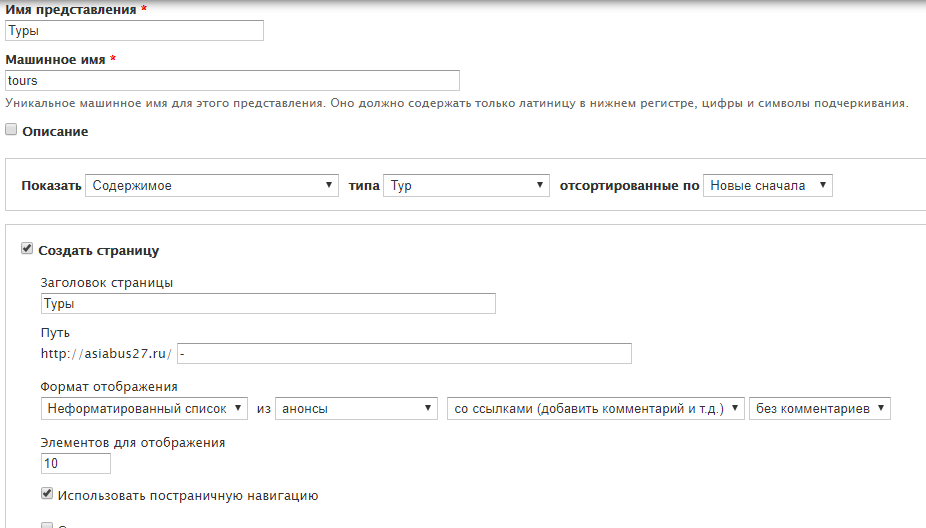


Рисунок ­ Окно создания нового представления

После кликаем продолжить и переходим на страницу редактирования представления. Здесь мы можем указать какие поля выводить, гибко настроить вывод полей, добавив им обертку, классы и т.д., настроить критерии фильтрации, сортировку, количество выводимых материалов и много другое. Для начала настроим формат, у нас уже указан тип вывода – неформатированный список, отображение полей материала. Для нашей темы лучше убрать обертку полей по умолчанию в настройках формата. В критериях фильтрации добавлен фильтр по типу материала, так что на странице будут выводится материалы необходимого типа.

Для того чтобы отображать нужные поля, нужно кликнуть по пункту добавить, справа от раздела – поля. Воспользовавшись поиском или просто выбрав нужный пункт (например, Изображение) добавляем его в вывод. Далее открывается окно настройки отображения, добавленного поля. В этом окне можно настроить метку, отключить ее и этого будет достаточно для этого поля.

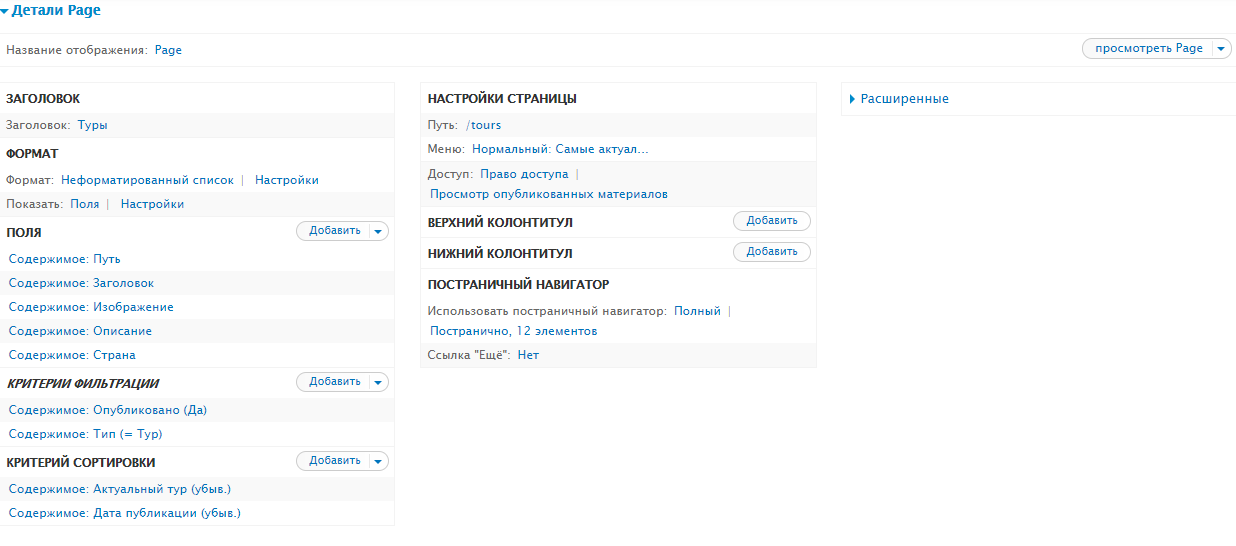


Рисунок – Страница настройки представления

Дальше добавляем следующие поля: заголовок; описание; страна.

Поле путь не добавлялось при создании материала, оно автоматически добавляется ко всем материалам модулем path. Ниже настроек можно кликнуть по кнопке обновить предпросмотр и увидеть уже готовое представление, но мы увидим не совсем то что нужно: картинки разных размеров, путь к материалу, который должен быть ссылкой на полную версию. По макету должно быть по 4 тура в ряду, круглое изображение и должны появляться эффекты при наведении (полоса снизу и замена цвета заголовка). Для того чтобы это настроить, нужно создать новый шаблон для вывода представления. Увидеть текущий шаблон можно в расширенных настройках в разделе другое, информация о теме. Здесь можно увидеть названия шаблона, который используется для вывода представления, полей, материалов и т.д.

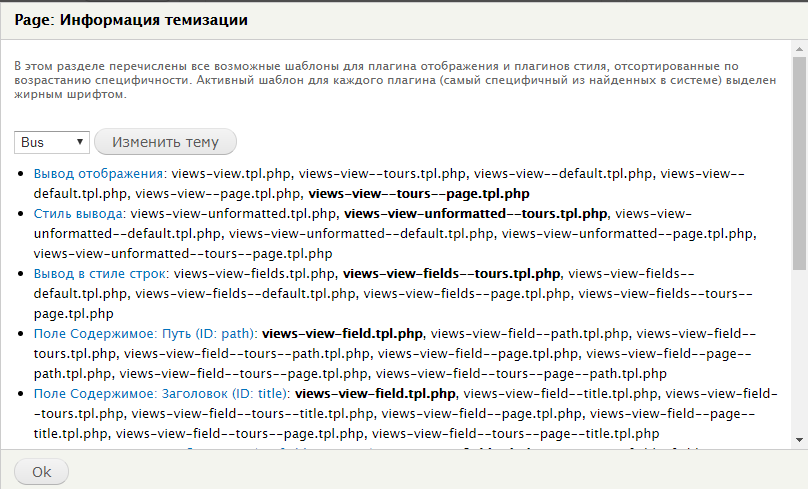


Рисунок – Информация темизации

Для нашего случая достаточно будет создать шаблоны для стиля вывода и вывода в стиле строк, где мы сверстаем вывод отдельного материала и структуру их вывода, соответственно. Жирным выделены названия текущих шаблонов. На рисунке выше уже созданы файлы views-view-unformatted—tours.tpl.php и views-view-fields—tours.tpl.php, эти названия более уникальны чем, названия шаблонов по умолчанию, поэтому представление будет выводится используя новые шаблоны.



Рисунок – Шаблон вывода в стиле строк

На рисунке выше показана разметка вывода одного материла типа – Тур. Данная разметка была скопирована из уже готовой верстки, только здесь статичный контент заменен динамичным при помощи php. Заменяются поля изображения, пути ссылки, заголовка и описания. Синтаксис php:

print $fields[‘имя поля’]->content;

Данная строка выводит содержимое поля, по его имени.

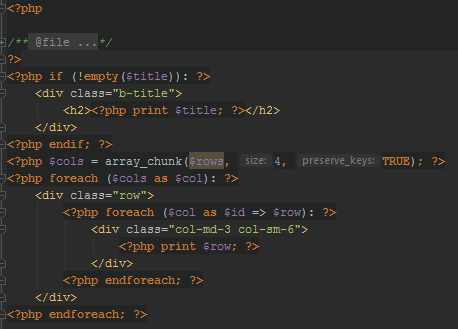


Рисунок ­ Шаблон стиля вывода

На данном изображении размечена структура вывода материалов. Массив $rows содержит все материалы типа – Тур. Для вывода будет использоваться сетка bootstrap, в каждой строке будет находится по 4 блока. Для этого делим массив всех материалов на 4, для того чтобы оборачивать новой строкой группу из четырех туров.

После сохранения последних двух шаблонов, можно будет увидеть результат пройдя на страницу туров.

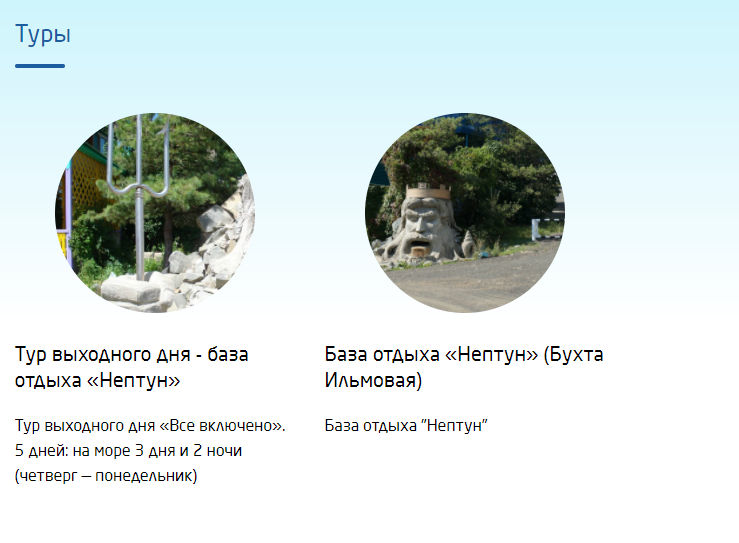


Рисунок – Страница туры

Представления для вывода автобусов и экскурсий создаются, почти идентично.

Создание слайдера

Иногда при разработке сайтов со сложной структурой, возникает желание объединить несколько полей типа материала так сказать в один пакет, при необходимости который мы смогли в один клик добавить необходимое количество этих пакетов. Для подобных случаев существует модуль Field collection (Коллекция полей), который добавляет новый тип поля, при создании материала. Коллекция полей представляет собой новый тип материала, но также является полем другого материала. В данном случае тип материала будет слайдер с одним полем (слайд) типа – коллекция полей. Поле слайд сможет иметь неограниченное количество значений и свои поля.

Создаем новый тип материала – Слайдер – добавляем одно поле, с названием – Слайды (field\_slides), количество значений устанавливаем неограниченное. Теперь, в пункте меню структура появился новый раздел – коллекция полей, а внутри field\_slides, к которому уже можно добавить поле для заголовка, текстового контента и изображения слайда, а также текстовое поле для ссылки.

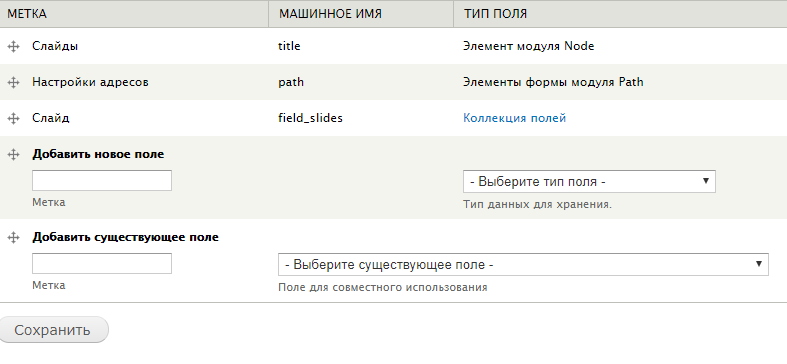


Рисунок – Добавление полей к новому материалу

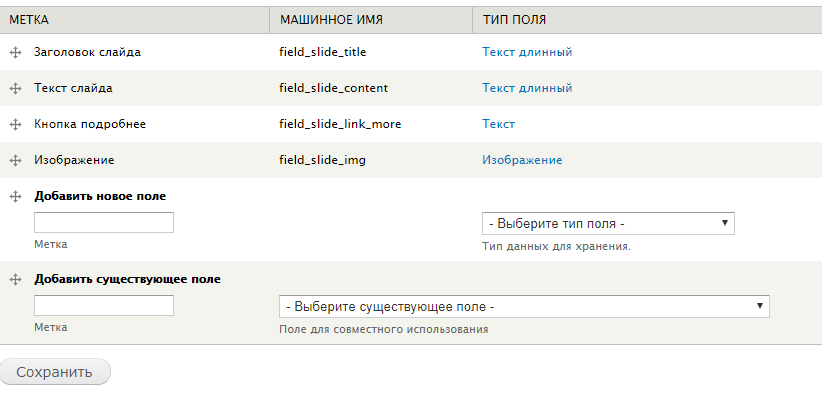


Рисунок – Добавление полей в коллекцию полей

Далее, необходимо создать несколько тестовых слайдов, затем настроить их дальнейшее отображение при помощи модуля view. Настройка представления отличается от предыдущей только тем, что в нем будут отображаться элементы коллекции полей и вместо страницы будет создан блок, который как уже говорилось, может быть отображен в любом месте сайта. Для создания слайдера можно использовать любой jquery-плагин, большое количество которых можно найти в интернете. На данном сайте это будет плагин Swiper.

После настройки шаблонов, слайдер нужно вывести на главной странице под списком стран. Есть несколько способов программного вывода блоков, в Drupal 7 это делается следующим образом:

Вывести только контент блока:

$block = module\_invoke($module\_name, 'block\_view', $block\_delta);

echo render($block['content']);

Вывести блок полностью с заголовком и контекстными ссылками:

$block = block\_load($module\_name, $block\_delta);

$block = \_block\_render\_blocks(array($block));

$block\_build = \_block\_get\_renderable\_array($block);

echo drupal\_render($block\_build);

Еще существует программный вывод блока с контекстными фильтрами, но для данного слайдера это не нужно.

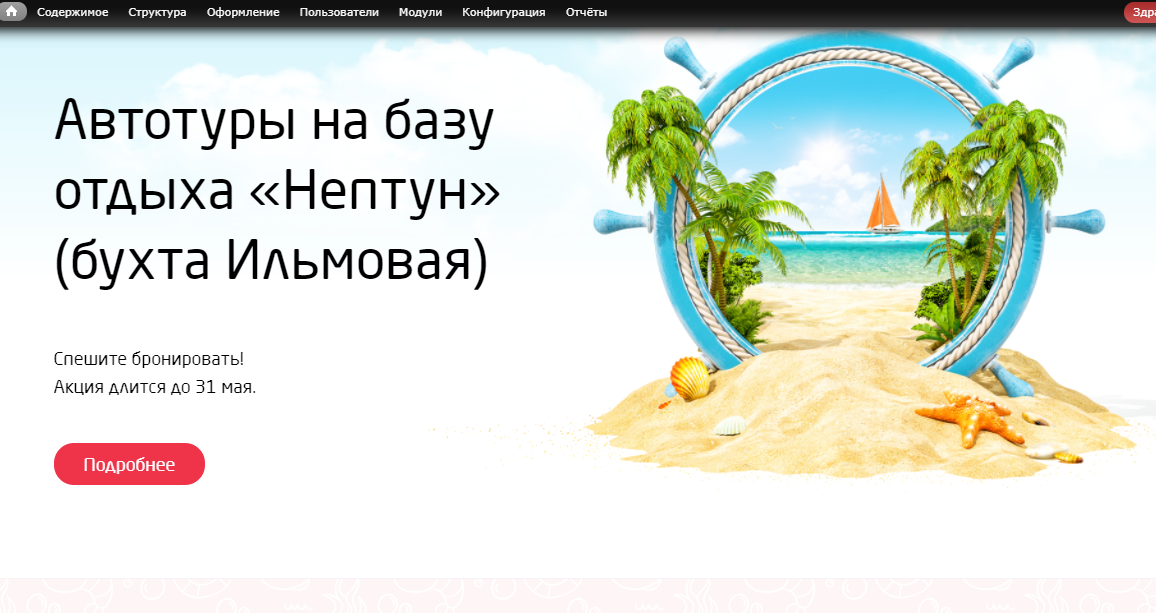


Рисунок – Слайдер на главной странице

Таксономия

Модуль taxonomy позволяет упорядочить материалы по категориям. На сайте asiabus словарем таксономии будет являться список стран, дял его создания нужно пройти в пункт меню Структура -> Таксономия -> Добавить словарь. Ввести название, машинное имя и (необязательно) описание, затем сохранить словарь.

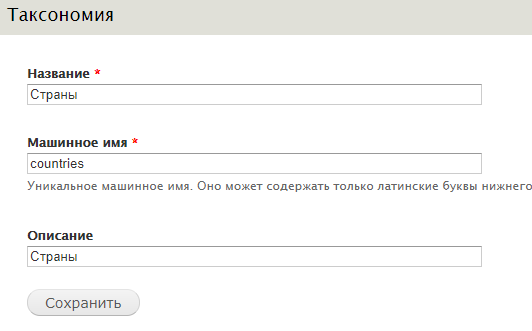


Рисунок – Создания словаря таксономии

Далее в словарь добавляются термины. Для словаря стран – терминами будут страны.

Во внутренних страницах каждого термина должны находиться туры для текущей страны, то есть необходимо связать тип материала и термин таксономии. Для этого при создании полей в материале, существует поле с типом – Ссылка на термин, где необходимо выбрать словарь, на термин которого будет ссылаться материал, а при создании нового материала типа Тур выбирать для него соответствующую страну.

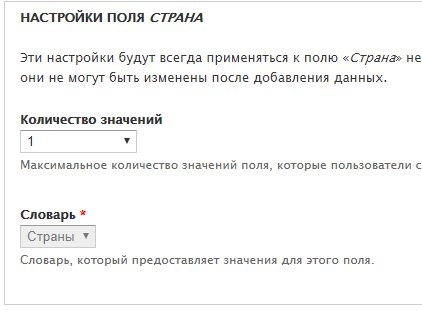


Рисунок – Настройка поля типа – Ссылка на термин

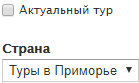


Рисунок – Выбор термина при создании материала

Теперь необходимо отобразить словарь таксономии, то есть список стран на странице, как в макете. Для этого будет использоваться модуль views, только в качестве выводимых значений будут термины таксономии.

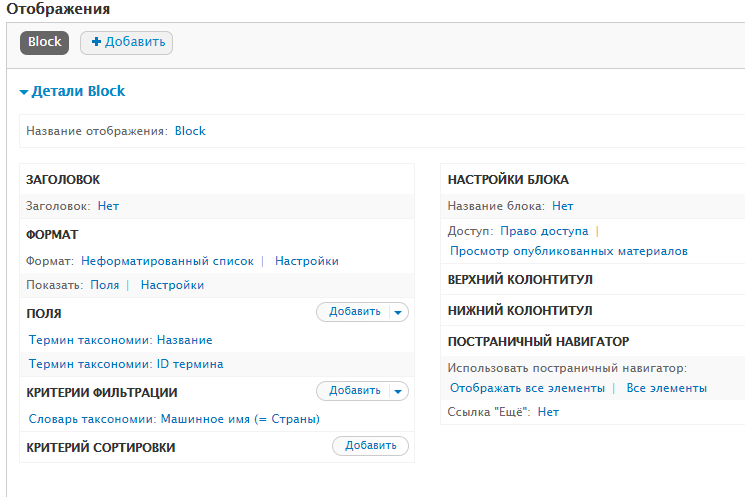


Рисунок – Создание блока страны

После программирования шаблонов блок можно вывести на странице.



Рисунок – Блок страны

Также нужно настроить внутренне отображение термина таксономии – вывод туров для страны. Здесь нужно просто создать новое представление и задать контекстный фильтр по термину таксономии. Шаблоны идентичны шаблонам представления с материалами типа – Тур.

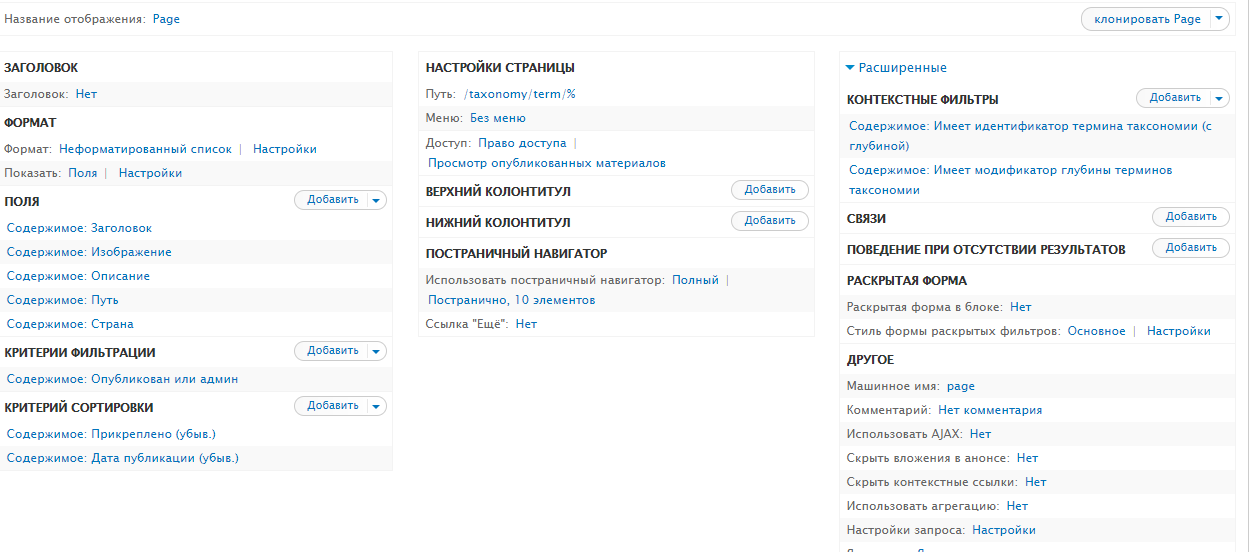


Рисунок – Настройка представления термина таксономии

«Хлебные крошки»

Навигационная цепочка (дублирующее меню, «хлебные крошки», англ. Breadcrumbs) – элемент навигации представляющий собой путь от корня сайта (главной страницы), который в данный момент просматривает пользователь. Для создания потребуется модуль Path Breadcrumbs. Переходим в пункт структура->path breadcrumbs->Добавить хлебные крошки. Для примера создадим страницу вывода туров.

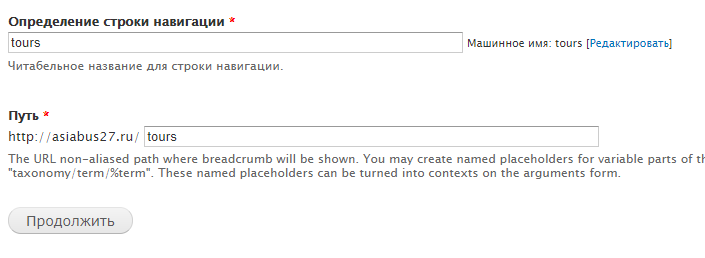


Рисунок – Создание хлебных крошек

Следующие пункты аргументы и правила выбора пропускаются. В настройках строки навигации выбираем отображение ссылки на главную страницу и добавляем в строку ссылку на текущую страницу (название – Туры, путь <none>). Путь ссылки текущей страницы необходимо оставить пустым.

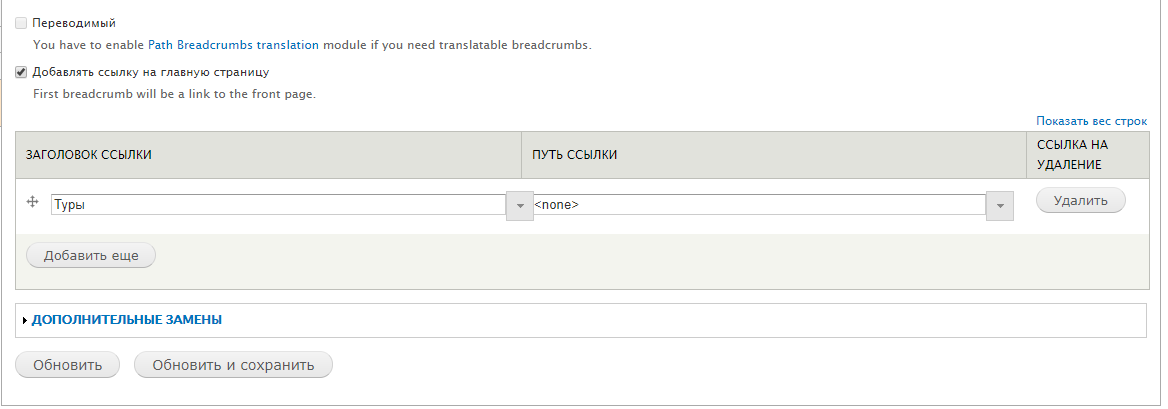


Рисунок – Настройка строки навигации

Далее настроим хлебные крошки на внутренние страницы туров. Путь ко всем материалам можно посмотреть в шаблонах в разделе Конфигурация->Поиск и метаданные->Синонимы URL. В данном случае это будет путь node/%node (для машины). Далее настраиваем аргумент строки

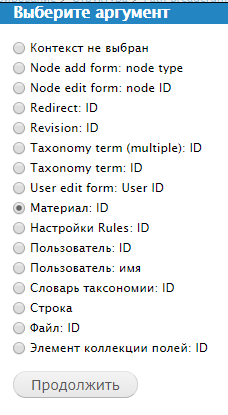


Рисунок – Выбор аргумента

В правилах выбора добавляем критерий по типу материала (тур).

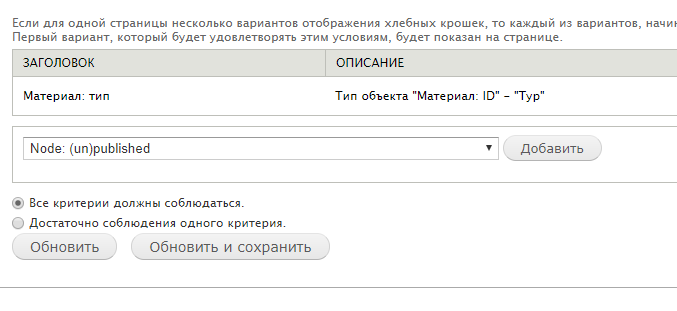


Рисунок – Настройка правил выбора

В настройках строки навигации также добавляем ссылку на главную страницу, ссылку на страницу туров и ссылку на текущий материал.

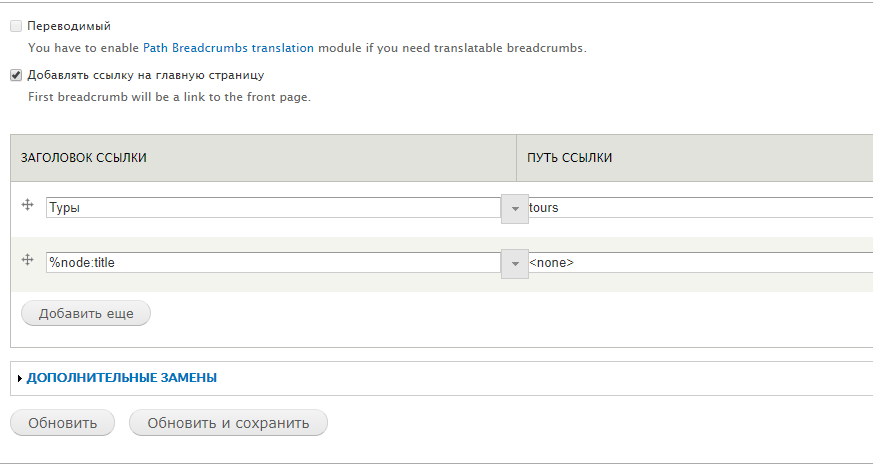


Рисунок – Настройка строки навигации материала

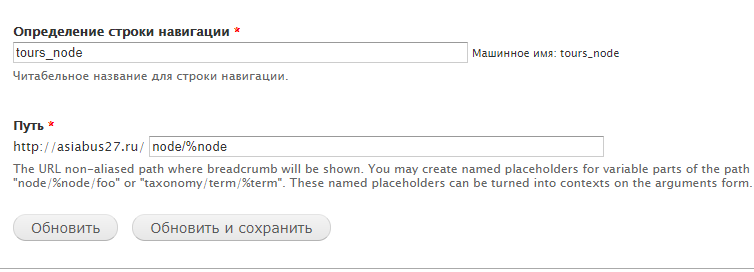


Рисунок – Создание строки навигации

Формы обратной связи (фос)

Для создания веб-форм нужен модуль webform, а также дополнительный модуль webform ajax для отправки форм без перезагрузки страницы.

Веб-формы являются новым типом материала и добавить их можно также, как и любой другой материал.

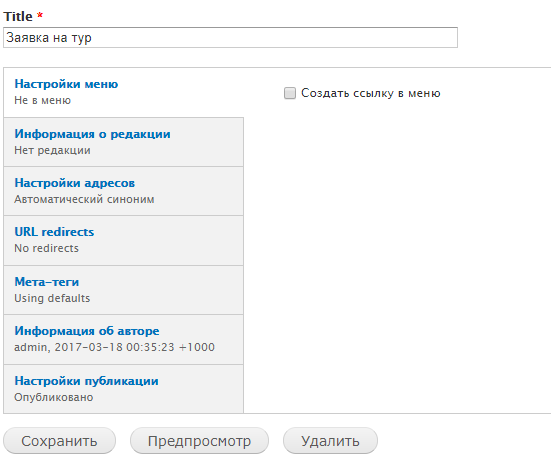


Рисунок – Создание новой формы

Далее добавляем компоненты формы (поля материала). Метка будет выводиться в html-элементе label, если включить его отображение, ключ формы вставиться в значение атрибута name, элемента формы. В заявке на тур есть 3 поля для ввода текста (Имя, Электронная почта, Телефон). Имя и телефон обязательны. В настройках компонента можно изменить его метку (label), заполнитель (атрибут placeholder), сделать поле обязательным (атрибут required), настроить класс компонента и класс обертки.

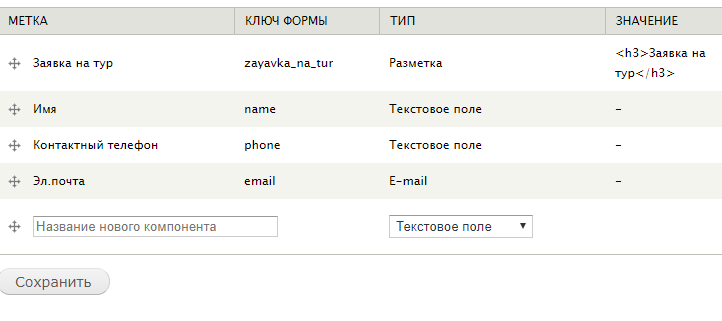


Рисунок – Форма заявки на тур

Далее, в настройках формы вводим текст сообщения о подтверждении успешной отправки формы, включаем режим ajax и вводим текст для кнопки отправки, разрешаем использовать форму в качестве блока, чтобы можно было вывести ее в любом месте сайта.

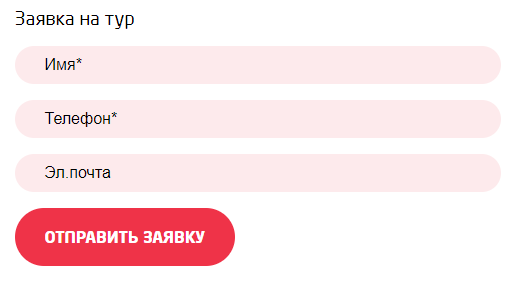


Рисунок – Форма заявки внутри страницы тура

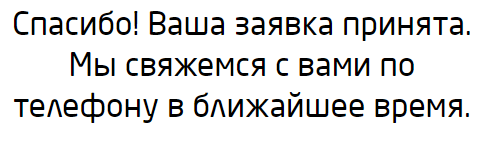


Рисунок – Ответ после успешной отправки формы

ЗАКЛЮЧЕНИЕ

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

Итан Браун. Веб-разработка с применением Node и Express. Полноценное использование стека JavaScript. = Web Development with Node and Express / Итан Браун;. - Санкт-Петербург: Питер, 2017. - 336 с.

Тодд Томлинсон. CMS Drupal 7: руководство по разработке системы управления веб-сайтом, 3-е издание = Pro Drupal 7 Development, 3rd Edition. — М.: [«Вильямс»](https://ru.wikipedia.org/w/index.php?title=%D0%92%D0%B8%D0%BB%D1%8C%D1%8F%D0%BC%D1%81_(%D0%B8%D0%B7%D0%B4%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D1%81%D1%82%D0%B2%D0%BE)&action=edit&redlink=1), 2011. — 560 с.

Мелансон Б., Нордин Д., Луиси Ж. Профессиональная разработка сайтов на Drupal 7. — СПб.: [«Питер»](https://ru.wikipedia.org/wiki/%D0%9F%D0%B8%D1%82%D0%B5%D1%80_(%D0%B8%D0%B7%D0%B4%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D1%81%D1%82%D0%B2%D0%BE)), 2013. — 688 с..

Алексей Черных. Drupal 7. — [«Эксмо»](https://ru.wikipedia.org/wiki/%D0%AD%D0%BA%D1%81%D0%BC%D0%BE_(%D0%B8%D0%B7%D0%B4%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D1%81%D1%82%D0%B2%D0%BE)), 2011. — С. 208.

Колисниченко Денис Николаевич. Drupal 7. Руководство пользователя. — [«Диалектика»](https://ru.wikipedia.org/w/index.php?title=%D0%94%D0%B8%D0%B0%D0%BB%D0%B5%D0%BA%D1%82%D0%B8%D0%BA%D0%B0_(%D0%B8%D0%B7%D0%B4%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D1%81%D1%82%D0%B2%D0%BE)&action=edit&redlink=1), 2011. — С. 256. — [ISBN 978-5-8459-1717-1](https://ru.wikipedia.org/wiki/%D0%A1%D0%BB%D1%83%D0%B6%D0%B5%D0%B1%D0%BD%D0%B0%D1%8F:%D0%98%D1%81%D1%82%D0%BE%D1%87%D0%BD%D0%B8%D0%BA%D0%B8_%D0%BA%D0%BD%D0%B8%D0%B3/9785845917171).

Ромашов В. CMS Drupal: система управления содержимым сайта (+CD с видеокурсом). — СПб.: [«Питер»](https://ru.wikipedia.org/wiki/%D0%9F%D0%B8%D1%82%D0%B5%D1%80_(%D0%B8%D0%B7%D0%B4%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D1%81%D1%82%D0%B2%D0%BE)), 2010. — С. 256.

Robert T. Douglass, Mike Little, Jared W. Smith «Building Online Communities with Drupal, phpBB, and WordPress». Apress, ISBN 1-59059-562-9

David Mercer «Drupal: Creating Blogs, Forums, Portals, and Community Websites»

John K. VanDyk, Matt Westgate «Pro Drupal Development»

В. Васвани. MySQL: использование и администрирование = MySQL Database Usage & Administration. — М.: «Питер», 2011. — 368 с.

Стив Суэринг, Тим Конверс, Джойс Парк. PHP и MySQL. Библия программиста, 2-е издание = PHP 6 and MySQL 6 Bible. — М.: «Диалектика», 2010. — 912 с.

Дмитрий Котеров, Алексей Костарев. PHP. — СПб.: «БХВ-Петербург», 2005. — С. 1120. — (В подлиннике).

Костарев А. Ф. PHP 5. — СПб.: «БХВ-Петербург», 2008. — С. 1104.

Мэтт Зандстра. PHP: объекты, шаблоны и методики программирования = PHP Objects, Patterns and Practice, Third Edition. — 3-е издание. — М.: «Вильямс», 2010. — С. 560.

Кристиан Дари, Эмилиан Баланеску. PHP и MySQL: создание интернет-магазина = Beginning PHP and MySQL E-Commerce: From Novice to Professional. — М.: «Вильямс», 2010.

Джейсон Ленгсторф. PHP и jQuery для профессионалов = Pro PHP and jQuery. — М.: «Вильямс», 2010. — С. 352.

Стив Суэринг, Тим Конверс, Джойс Парк. PHP и MySQL = PHP 6 and MySQL 6 Bible. — 2-е издание. — М.: «Диалектика», 2010. — С. 912. — (Библия программиста).

Квентин Зервас. Web 2.0: создание приложений на PHP = Practical Web 2.0 Applications with PHP. — М.: «Вильямс», 2009. — С. 544.

Кузнецов Максим, Симдянов Игорь. PHP 5/6. — СПб.: «БХВ-Петербург», 2009. — С. 1024.

Э. Фримен, Э. Фримен. Изучаем HTML, XHTML и CSS = Head First HTML with CSS & XHTML. — П.: «Питер», 2010. — 656 с.

Эд Титтел, Джефф Ноубл. HTML, XHTML и CSS для чайников, 7-е издание = HTML, XHTML & CSS For Dummies, 7th Edition. — М.: «Диалектика», 2011. — 400 с.

Питер Лабберс, Брайан Олберс, Фрэнк Салим. HTML5 для профессионалов: мощные инструменты для разработки современных веб-приложений = Pro HTML5 Programming: Powerful APIs for Richer Internet Application Development. — М.: «Вильямс», 2011. — 272 с.

Стивен Шафер. HTML, XHTML и CSS. Библия пользователя, 5-е издание = HTML, XHTML, and CSS Bible, 5th Edition. — М.: «Диалектика», 2010. — 656 с.

Фримен Эрик, Фримен Элизабет. Изучаем HTML, XHTML и CSS = Head First HTML with CSS & XHTML. — 1-е изд. — М.: «Питер», 2010. — С. 656.

Эд Титтел, Джефф Ноубл. HTML, XHTML и CSS для чайников, 7-е издание = HTML, XHTML & CSS For Dummies, 7th Edition. — М.: «Диалектика», 2011. — 400 с.

Стивен Шафер. HTML, XHTML и CSS. Библия пользователя, 5-е издание = HTML, XHTML, and CSS Bible, 5th Edition. — М.: «Диалектика», 2011. — 656 с.

Энди Бадд, Камерон Молл, Саймон Коллизон. CSS: профессиональное применение Web-стандартов = CSS Mastery: Advanced Web Standards Solutions. — М.: «Вильямс», 2008. — 272 с.

Кристофер Шмитт. CSS. Рецепты программирования = CSS. Cookbook. — СПб.: БХВ-Петербург, 2007. — 592 с.

Эрик А. Мейер. CSS-каскадные таблицы стилей: подробное руководство = Cascading Style Sheets: The definitive Guide. — М.: Символ, 2006. — 576 с.

Адам Фримен. jQuery для профессионалов = Pro jQuery. — М.: «Вильямс», 2012. — 960 с.

Джейсон Ленгсторф. PHP и jQuery для профессионалов = Pro PHP and jQuery. — М.: «Вильямс», 2010. — С. 352. — ISBN 978-5-8459-1693-8.

Самков Г. jQuery. Сборник рецептов. — СПб.: БХВ-Петербург, 2010. — С. 416.

Питер Бойер. Photoshop CS6 для чайников = Photoshop CS6 For Dummies. — М. : «Диалектика», 2012. — 448 с.

https://www.drupal.org/

http://gulpjs.com/

https://yandex.ru/referats/?ncrnd=3492

http://getbootstrap.com/

https://docs.emmet.io/

https://htmlacademy.ru/

http://lesscss.org/

https://github.com/

https://git-scm.com/

https://yarnpkg.com/lang/en/

https://www.npmjs.com/

https://bower.io/

https://www.jetbrains.com/phpstorm/

https://www.sublimetext.com/

https://conemu.github.io/

https://photoshop-master.ru/

https://fonts.google.com/