
Доработка интерфейса

Доработка интерфейса системы поддержки издательской деятельности OJS в соответствии с корпоративным стилем государственного университета 'Дубна' для сетевого научного издания 'Системный анализ в науке и образовании'

Кунгуров Макар, Геоня Татьяна, Кузнецов Михаил,

Авдеева Эльвира

2021-05-05



Оглавление

Инструменты и технологии.	2
Сервер Apache	2
Архитектура Apache	2
Ядро	2
Модульная система	2
Достоинства и недостатки Apache	2
OJS	3
Назначение OJS	4
Архитектура OJS	4
Достоинства и недостатки OJS	4
Плагин bootstrap3	5
Проделанная работа	5
Развернут сайт	5
Создание собственной темы bootstrap3	7
Структура плагина	7
Установка плагина	7
Шаблоны tpl	7
Блок статей	8
Блок описания	8
Блок навигации	9
Блок подвала	9
Страница архива	9
Блок журнала в архиве журналов	9
Переменные(локализация) для страницы добавлять в файл	11
Пример переноса переменных на другую страницу	11
Кастомизация стилей	12
Выводы	13
Дополнительные плагины	15
Документация на русском языке	15

Инструменты и технологии.

Сервер Apache

Apache – это свободное программное обеспечение для размещения веб-сервера. Он хорошо показывает себя в работе с масштабными проектами, поэтому заслуженно считается одним из самых популярных веб-серверов. Кроме того, Apache очень гибок в плане настройки, что даёт возможность реализовать все особенности размещаемого веб-ресурса.

Архитектура Apache

Apache состоит из ядра и динамической модульной системы. Параметры системы изменяются с помощью конфигурационных файлов.

Ядро

Ядро Apache разработано Apache Software Foundation на языке C. Основные функции — обработка конфигурационных файлов, протокол HTTP/HTTPS и загрузка модулей. Ядро может работать без модулей, но будет иметь ограниченный функционал.

Модульная система

Модуль – отдельный файл, подключение которого расширяет изначальный функционал ядра. Они могут включаться в состав ПО при первоначальной установке или подгружаться позже через изменение конфигурационного файла.

Большинство из них отвечает за определенный аспект обработки клиентского запроса – поддержку различных языков программирования, безопасность, кэширование, аутентификацию и т.д. Таким образом, большая задача разбивается на несколько фаз, каждую из которых решает отдельный, узкоспециализированный модуль.

Для Apache существует больше 500 модулей. Многие популярные веб-приложения сразу выпускаются в виде модуля к Apache. Например, ISPmanager и VDSmanager.

Достоинства и недостатки Apache

Плюсы

- Доступность. Это программное обеспечение с открытым исходным кодом. Значит, его может бесплатно использовать или модифицировать любой желающий. Разработчики по всему миру создают конфигурации и модули веб-сервера для своих специфических нужд. По этой же причине Apache регулярно получает полезные дополнения, расширяющие его базовый функционал.
- Гибкость настройки. Apache использует несколько конфигурационных файлов для управления веб-сервером. Это позволяет настроить ПО под узконаправленные задачи.
- Функциональность. У Apache динамическая модульная структура. Можно быстро подключать дополнительный функционал в виде скачиваемых модулей, даже без обращения к внешним источникам. Это позволяет решать целый комплекс важнейших задач в области безопасности, кэширования, редактирования URL, распределения нагрузки. Благодаря гибридным модулям MPM, Apache может одинаково успешно обслуживать статический и динамический контент. Есть возможность оперативно отключать ненужные модули и ускорять работу веб-сервера.
- Кроссплатформенность. Apache работает как на Windows, так и на всех Unix-подобных системах. Администрирование веб-сервером не имеет серьёзных отличий на разных ОС. Индивидуален только процесс установки и расположение директорий с файлами программы.
- Совместимость. Apache работает на базе скриптовых или веб-ориентированных языков (PHP, Python, Tcl, Ruby, Perl, ASP), что делает его совместимым с самым широким спектром баз данных и серверного ПО. Многие веб-приложения и инструменты сразу выходят со средствами запуска из-под Apache в виде PHP-модуля. Веб-сервер, поддерживает технологии FastCGI и CGI, позволяющие пользоваться программными продуктами на объектно-ориентированных языках Java, sh, C, C++.
- Масштабируемость. Подходит для веб-ресурсов любого масштаба. Apache хорошо работает как на одностраничном сайте (лендинге), так и на многостраничном сайте с ежедневной аудиторией в десятки тысяч посетителей.

OJS

Open Journal Systems (OJS, «Открытые журнальные системы») — открытое программное обеспечение для организации рецензируемых научных изданий, разработанное некоммерческим исследовательским проектом Public Knowledge Project. Распространяется по лицензии GNU General Public License.

Назначение OJS

Система OJS предназначена для создания рецензируемых электронных журналов с открытым доступом и позволяет не только публиковать статьи в интернете, но и организовывать весь рабочий процесс издательского дела: приём, рецензирование и каталогизирование статей. Подача статей производится через интернет.

В журналах OJS пользователи делятся на группы с различными привилегиями: менеджер издания, редактор, рецензент, автор, читатель и другие. OJS включает в себя расширение для распространения журнала по подписке. Редакторы журнала настраивают требования к статьям, секции, по которым принимаются статьи, процесс рецензирования и многое другое.

Архитектура OJS

Система OJS написана на PHP и может быть запущена на любом веб-сервере с поддержкой данного интерпретатора; в качестве базы данных используется MySQL или PostgreSQL.

Модульная система OJS обладает модульной структурой и имеет возможность подключения плагинов, для расширения функционала, аналогично многим открытым продуктам, разрабатываемым сообществами разработчиков, таких как WordPress. Существуют плагины, позволяющие индексировать содержимое журнала в Google Scholar и PubMed Central. Плагин подписки реализует поддержку стандартов RSS и Atom. Open Journal Systems соответствует стандартам проекта LOCKSS, что позволяет безопасно собирать, хранить и предоставлять доступ ко всем статьям журнала в долгосрочной перспективе. OJS поддерживает идентификаторы цифровых объектов, что позволяет регистрировать статьи в таких агентствах, как CrossRef, Multilingual European DOI Registration Agency и DataCite.

Дополнительные инструменты Для вовлечения читателей в процесс создания журнала сообществом Public Knowledge Project был разработан набор инструментов Reading Tools, предоставляющий доступ к смежным исследованиям, тематическим новостям, законодательным актам и другим ресурсам в открытых базах данных.

Достоинства и недостатки OJS

- OJS является свободно распространяемым программным обеспечением с открытым кодом. Т.е. Вы не зависите от разработчиков и навязываемых лицензионных договоров.

- OJS изначально создавалась как специализированная издательская платформа. Она поддерживает обмен данными с международными реферативными базами данных и библиотеками.
- Имеет низкую стоимость разработки и поддержки сайтов на OJS.

Плагин bootstrap3

Плагин bootstrap3 - встроенный плагин-тема, разработанный сообществом OJS, для упрощения работы с темами и стилями в OJS.

Проделанная работа

Исходя из заданной цели, благодаря преподавателям была создана команда из 5 студентов Университета “Дубна”. Каждый выполнял свои роли и отвечал за назначенную сферу ответственности.

Список совершенной работы:

- Развернут сайт по адресу sanse.uni-dubna.ru
- Полностью изменен стандартный вид и тема оформления сайта
 - Изменены структуры страниц сайта
- Добавлена мультиязычная функциональность
- Добавлены дополнительные плагины, упрощающие пользовательское взаимодействие с сайтом

Развернут сайт

Сайт развернут на домене sanse.uni-dubna.ru при помощи вышеописанных технологий сервера apache и системы OJS. В процессе разработки были попытки использовать контейнеризацию системы Docker, но впоследствии идея оказалась бессмысленной. При помощи стандартных инструментов студенты развернули сайт, добившись его полного функционирования. Развертка применяла следующие рекомендуемые настройки:

1. Выделите базу данных для OJS; используйте уникальные учетные данные для доступа к ней. Настройте базу данных для автоматического резервного копирования на регулярной основе. Выполнять резервное копирование вручную при обновлении или выполнении сопровождения.

2. Настройте OJS (config.inc.php) для использования хэширования SHA1 вместо MD5.
3. Настройте OJS (config.inc.php), чтобы использовать force_ssd_login, чтобы аутентифицированные пользователи обменивались данными с сервером по протоколу HTTPS.
4. Установите OJS, чтобы каталог файлов НЕ являлся подкаталогом установки OJS и не мог быть доступен напрямую через Интернет. Как можно больше ограничьте права доступа к файлам. Автоматизированный резервные копии этого каталога должны быть синхронизированы с резервным копированием базы данных. Установка производилась по следующей инструкции:
5. Разархивируйте архив OJS в нужное место в вашей сети каталога документов.
6. Сделайте следующие файлы и каталоги (и их содержимое) доступным для записи (то есть путем изменения владельца или разрешения с помощью команды chown или chmod):
 - config.inc.php (необязательный параметр - если не доступен для записи, вам будет предложено вручную перезаписать этот файл во время установки)
 - public
 - cache
 - cache/t_cache
 - cache/t_config
 - cache/t_compile
 - cache/_db
3. Создайте каталог для хранения загруженных файлов (файлов отправки и т.д.) и сделайте этот каталог доступным для записи. Рекомендуется, чтобы эта директория была размещена в не доступном их Интернету месте (или иначе защищена от прямого доступа, например, посредством правил .htaccess).
4. Откройте веб-браузер и введите `http://yourdomain.com/path/to/ojs/` . Следуйте инструкциям на экране. В качестве альтернативы, можно использовать установщик командной строки запустив команду “`php tools/install.php`» из вашего каталога OJS. (Примечание: с установщиком CLI вам может понадобиться chown/chmod общедоступные каталоги загруженных файлов после установки, если пользователь Apache отличается от пользователя, запускающего инструмент.)
5. Рекомендуемые дополнительные действия после установки:
 - Просмотрите config.inc.php для дополнительных настроек конфигурации.
 - Просмотрите документ часто задаваемых вопросов для часто задаваемых вопросов технического характера и вопросов конфигурации сервера.

Создание собственной темы bootstrap3

Структура плагина

файлы которые позволяют кастомизировать стили bootstrap 3 находятся в папке:

```
ojs-site/plugins/themes/bootstrap3/bootstrap-themes/
```

шаблоны страниц находятся по ссылке:

```
ojs-site/plugins/themes/bootstrap3/templates
```

файлы для локализации находятся по ссылке:

```
sanse-test/plugins/themes/bootstrap3/locale
```

Пример вызова шаблона страницы на примере главной страницы

Когда мы заходим на главную страницу, используется шаблон по ссылке

```
plugins/themes/bootstrap3/templates/frontend/pages/indexJournal.tpl
```

данный шаблон выводится пользователю из документа по ссылке:

```
site/sanse-test/pages/index/IndexHandler.inc.php
```

Там так-же объявляются переменные, которые используются шаблоном

Установка плагина

Чтобы подключить данный плагин, достаточно его распаковать в папку:

```
ojs-site/plugins/themes
```

Важно отметить, что версию плагина надо выбирать относительно версии ojs.

git репозиторий данного плагина:

```
https://github.com/NateWr/bootstrap3
```

Шаблоны tpl

В данном пункте будут показаны не все шаблоны, так как они могут дополняться и удаляться, в зависимости от версии OJS. Однако этих шаблонов будет достаточно, чтобы в общих чертах понять их структуру вызова.

Блок статей

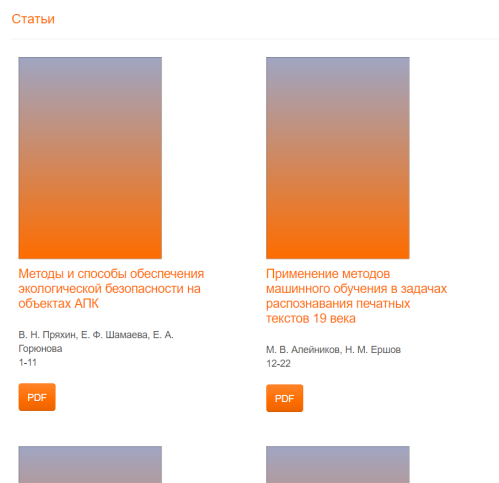


Рис. 1: Блок статей

```
1 {include file="frontend/objects/issue_toc.tpl"}
```

Вызывается в шаблоне:

```
bootstrap3/templates/frontend/pages/indexJournal.tpl
```

Блок описания

встроен в шаблон:

```
1 {* Issue introduction area above articles *}
2 "<div>...</div>"
```

Вызывается в шаблоне:

```
bootstrap3/templates/frontend/pages/indexJournal.tpl
```

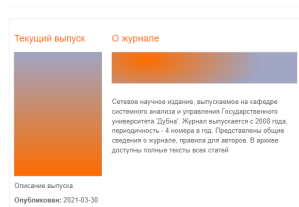


Рис. 2: Блок описания

Блок навигации

встроен в шаблон по комментарию:

```
1 {include file="frontend/components/header.tpl" pageTitleTranslated=
  $currentJournal->getLocalizedName()}
```

Вызывается в шаблоне:

```
bootstrap3/templates/frontend/pages/indexJournal.tpl
```

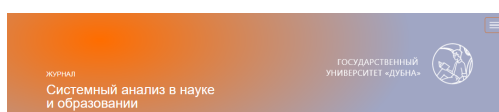


Рис. 3: Навигация

Блок подвала

Вызывается в шаблоне:

```
bootstrap3/templates/frontend/pages/indexJournal.tpl
```

встроен в шаблон по комментарию:

```
1 {include file="frontend/components/footer.tpl"}
```

Страница архива

```
bootstrap3/templates/frontend/pages/issueArchive.tpl
```

Блок журнала в архиве журналов

Встроен в шаблон по комментарию:

```
1 {include file="frontend/objects/issue_summary.tpl" home=true}
```

Вызывается в шаблоне:

```
frontend/pages/issueArchive.tpl
```

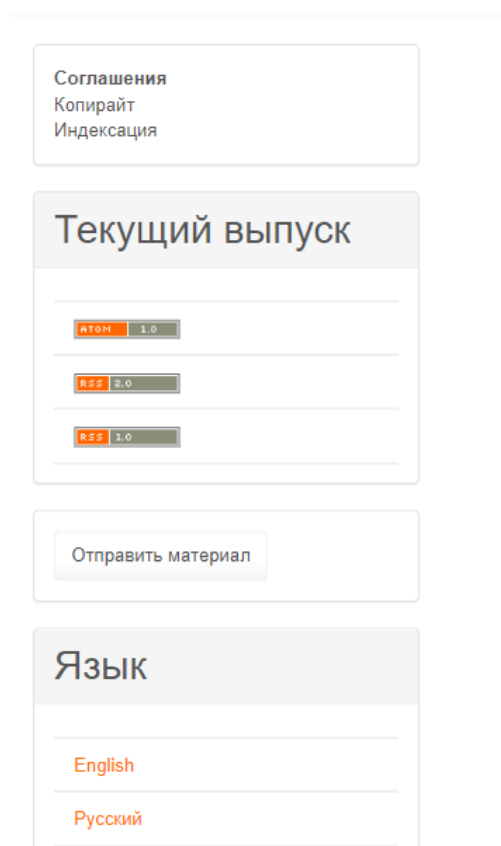


Рис. 4: Подвал-1

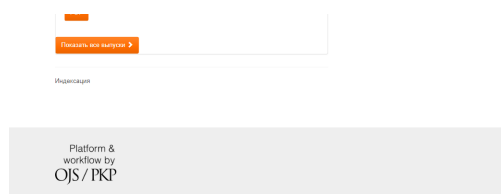


Рис. 5: Подвал-2

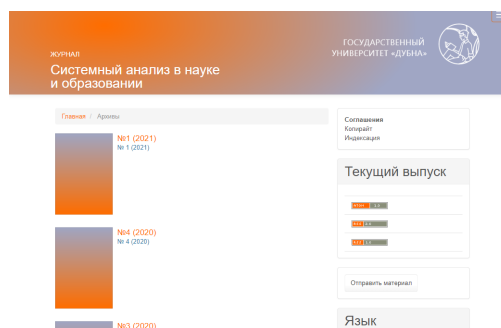


Рис. 6: Страница архива

Переменные(локализация) для страницы добавлять в файл

sanse-test/lib/pkp/locale/ru_RU/user.po Пример создания локальной переменной для требуемого языка:

```
1 msgid "api.my.1"
2 msgstr "journal"
```

Пример вызова в шаблоне:

```
1 {translate key="api.my.1"}
```

данная команда будет получать текстовую переменную journal

Чтобы корректно отображалась в разных языках, надо добавить перевод и в другие папки по языкам

Пример переноса переменных на другую страницу

по ссылке:

site/sanse-test/pages/index

найти в документ IndexHandler.inc.php

```
1 $templateMgr->display('frontend/pages/indexJournal.tpl');
```

это запуск шаблона главной страницы, если журнал существует и публичен по ссылке:

site/sanse-test/pages/issue/

найти в IssueHandler.inc.php

```
1 $templateMgr->display('frontend/pages/issueArchive.tpl');
```

Это запуск шаблона архива с журналами

Например, надо переместить все выпуски, которые передаются по ссылке архив, на главную страницу. Готовый проект этого не предусматривает. Но мы знаем, что перед тем, как в архив попадают журналы, они вызываются из базы данных.

за выпуски отвечает переменная issues. Это мы знаем из шаблона

bootstrap3/templates/frontend/pages/issueArchive.tpl

Смотрим команды которые надо исполнить, чтобы достать эти данные

```
1 $this->setupTemplate($request);
2
3 $page = isset($args[0]) ? (int) $args[0] : 1;
4
5 $templateMgr = TemplateManager::getManager($request);
6
7 $context = $request->getContext();
8 $count = $context->getData('itemsPerPage') ? $context->getData('
    itemsPerPage') : Config::getVar('interface', 'items_per_page');
9
10 $offset = $page > 1 ? ($page - 1) * $count : 0;
11
12 $params = array(
13
14     'contextId' => $context->getId(),
15     'orderBy' => 'seq',
16     'orderDirection' => 'ASC',
17     'count' => 3,
18     'offset' => $offset,
19     'isPublished' => true,
20 );
21
22 $issues = iterator_to_array(Services::get('issue')->getMany($params
    ));
```

Их мы нашли в файле IssueHandler.inc.php и перенесли в IndexHandler.inc.php

далее надо добавить в менеджер переменную, и уже обработать в запрашиваемом шаблоне

```
1 $templateMgr->assign(array('issues' => $issues,))
```

теперь она будет доступна в шаблоне tpl и ее можно смело использовать по образу того, как она используется в шаблоне:

```
bootstrap3/templates/frontend/pages/issueArchive.tpl
```

Мы ограничили их число ранее, чтобы не бросалось в глаза. За это отвечала переменная count. В итоге, развернув журналы по образу другого шаблона, мы можем увидеть на главной странице нужные журналы.

Кастомизация стилей

файлы которые позволяют кастомизировать стили bootstrap 3 находятся в папке:

```
bootstrap3/bootstrap-themes/
```

Мы кстомизировали тему cerulean

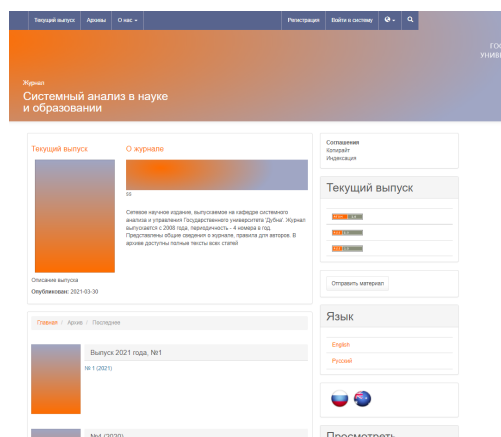


Рис. 7: Общий вид страницы

Стили находятся в документе по ссылке:

```
bootstrap3/bootstrap-themes/cerulean/bootswatch.less
```

Переменные для стилей находятся по ссылке

```
bootstrap3/bootstrap-themes/cerulean/variables.less
```

В данных файлах изначально добавлены только стили, которые изменяют уже готовые стили bootstrap 3. Их можно использовать, для кастомизации и добавления своих стилей. Из данных файлов компилируется файл стилей, который после используется на всех страницах сайта.

Выводы

В результате работы с плагином ojs 3 for bootstrap 3 была изучена общая структура сайта, ссылочной системы, стилизации и локализации OJS 3. Кроме того, проведенная работа закрепила навыки в css, js, разворачивании локального сервера.

Была проведена работа с навигацией. Несколько уровней меню были объединены. Также в меню был добавлен выбор языка. Меню было стилизовано по шаблону данному университетом.

На главную страницу был добавлен архив последних выпусков
вставлен блок отвечающий за плагины.

Была проведена кастомизация темы bootstrap 3.



Рис. 8: Шаблон



Рис. 9: Архив

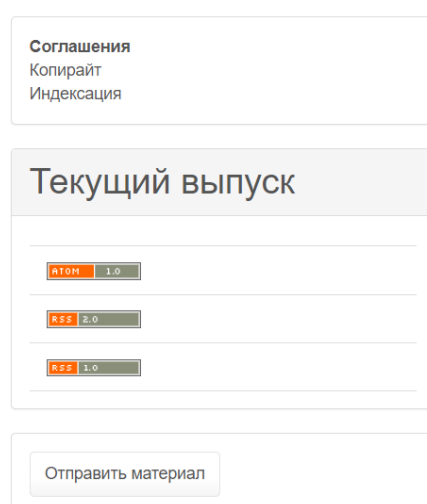


Рис. 10: Блок отвечающий за плагины

Добавленные переменные типа `string` были локализованы.

Все вышеперечисленное нормально отображается на различных экранах, в том числе и на телефонах.

Дополнительные плагины

1. Плагин создания собственных страниц;
2. Плагин цитирования в различных форматах;
3. Плагин облака тэгов;
4. Модуль метаданных `dublin core`;
5. PDF просмотрщик;
6. Модуль менеджер пользовательских блоков;
7. Модуль быстрой отправки.

Документация на русском языке

Дополнительная документация на русском языке может быть найдена [здесь](#)