

Интеграция CMS системы WordPress и Framework Yii2

Эрдман Александр Алексеевич

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

В статье рассмотрен процесс интеграции CMS системы WordPress и Framework Yii2. Интеграция проводится посредством встроенных возможностей двух систем. Результатом исследования является возможность настройки сайта, разработанного на Yii2, с помощью возможностей обеих систем.

Ключевые слова: Yii2, WordPress, интеграция

Integration of the WordPress CMS system and the Yii2 Framework

Erdman Alexander Alekseevich

Sholom-Aleichem Priamursky State University

Student

Abstract

The article discusses the integration process of the WordPress CMS system and the Yii2 Framework. Integration is carried out through the built-in capabilities of the two systems. The result of the study is the possibility of configuring a site developed on Yii2 using the capabilities of both systems.

Keywords: Yii2, WordPress, integration

1 Введение

1.1 Актуальность

В современных реалиях обыденным делом становится создание сайта под различные нужды, начиная от создания простых сайтов блогов, заканчивая интернет-магазинами. Наравне с потребностью в сайтах, начинает активно развиваться потребность в уникальности данного сайта, из-за чего разработчики всё чаще прибегают к более гибким системам создания сайта, а именно к полноценным PHP-framework системам, популярной из которых является Yii2, а также Laravel, CodeIgniter, Symfony и так далее. Но это выбор специалистов-разработчиков, который не всегда совпадает с выбором заказчика сайта. Зачастую заказчики, перед тем как обращаться к специалистам, проявляют интерес к принципу создания сайта, то есть при помощи чего делаются современные сайты, и какая система является самой популярной, а также более простой и как следствие более экономичной в затратах на разработку. По этой причине их выбор падает на CMS системы, самой популярной из которых является WordPress. Но данная CMS система

не даёт полноценной гибкости в сайтостроение, а также имеет уязвимости в безопасности. Этому мнению придерживаются опытные web-разработчики. Но часто мнение заказчика может не совпадать с мнением разработчика, что в свою очередь образует проблему. Решением данной проблемы является интеграция двух систем. В статье в качестве примера решения данной проблемы, а именно интеграции, будут рассмотрены CMS WordPress и PHP-framework.

1.2 Обзор исследований

Ю.Д. Романов и Т.Н. Копышева провели интеграцию модуля CMS Joomla с программным обеспечением «1С: Предприятие» для обработки информации на сайте интернет-магазина [1]. А.А. Кузнецов и А.П. Толстобров разработали компонент интеграции CMS Joomla и платёжной системы PayPal [2]. А.И. Вегнер исследовал системы управления студенческими проектами и технологии интеграции внутренних и внешних систем [3]. М.В. Пургина, А.С. Добрынин и Р.С. Койнов рассмотрели проблему интеграции отдельных АРМов в единую корпоративную среду в рамках Web-технологий [4]. В.А. Казакевич и В.Н. Бабешко рассмотрели особенности использования фреймворков, основанных на архитектуре оригинальной MVC [5]. М.Р. Эмомалиев и А.А. Попов проанализировал возможности популярных фреймворков: Symfony, Yii и Laravel [6].

1.3 Цель исследования

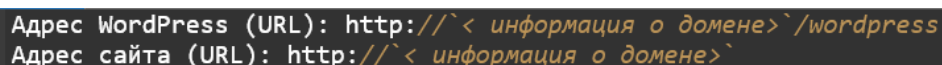
Целью исследования является интеграция CMS системы WordPress в систему Yii2 для удобства редактирования и визуальной настройки сайта.

2. Материалы и методы

Интегрируемыми системами выступают CMS WordPress и framework Yii2. Интеграция будет проходить посредством встроенных возможностей фреймворка Yii2.

3 Результаты и обсуждение

Интеграция двух данных систем начинается с установки и распаковки WordPress в отдельной папке, расположенной в корневой папки web-сайта. Это позволяет избежать нежелательного автообновления WordPress, которое будет перезаписывать отредактированные файлы, в частности сценарии входа. Далее необходимо удостовериться, что общие настройки WordPress настроены следующим образом (рис. 1).



```
Адрес WordPress (URL): http://^< информация о домене>/wordpress
Адрес сайта (URL): http://^< информация о домене>
```

Рис. 1. Вид адресов (URL) WordPress и сайта

Вышеприведенные настройки указывают WordPress, куда установить свои ссылки для запуска через входной скрипт Yii2, то есть пути для медиафайлов, таблиц, js и стилей в папку '/wordpress'. После настройки

WordPress, нужно настроить скрипт входа Yii2. Для этого нужно в сценарий ввода прописать следующий код (рис. 2)

```
1 define('WP_USE_THEMES', true);
2 $wp_did_header = true;
3 require_once('wordpress/wp-load.php');
4
5 require_once(dirname(__FILE__) . '/../protected/components/ExceptionHandler.php');
6 $router = new ExceptionHandler();
7
8 ....
9
10 require_once($yii);
11 Yii::createWebApplication($config)->run();
```

Рис. 2. Настройка скрипта входа в Yii2

После первичной настройки необходимо в web-папке сайта удалить файл 'index.php' и создать новый файл '.htaccess' со следующим содержимым (рис. 3). Для очистки URL-адресов необходимо для компонента 'urlManager' в свойстве 'showScriptName' прописать значение 'false'.

```
1 RewriteEngine on
2
3 # if a directory or a file exists, use it directly
4 RewriteCond %{REQUEST_FILENAME} !-f
5 RewriteCond %{REQUEST_FILENAME} !-d
6
7 # otherwise forward it to index.php
8 RewriteRule . index.php
```

Рис. 3. Создание .htaccess файла

Также необходимо создать собственный обработчик исключений WordPress (рис. 4). Обработчик позволяет обрабатывать ошибки 404, используя WordPress вместо страницы 404 Yii2. При наличии ошибки 404, данная ошибка будет отображаться в макете WordPress. Для того, чтобы Yii2 мог корректно взаимодействовать с WordPress, нужно создать контроллер, который позволит обрабатывать представления WordPress (рис. 5). Обязательно нужно отключить макеты Yii2, так как WordPress будет обрабатывать свои собственные макеты. Использование '\$this->render, Yii::app()->clientScript' позволяет файлам 'css/js' загружаться в заголовок WordPress автоматически после того, как WordPress отобразит своё содержимое. Для создания входящей точки в WPController, необходимо создать файл 'index.php', который будет располагаться в следующей директории 'view/wp/index.php' (рис. 6). После выполнения вышеописанных действий при переходе по основному URL-адресу должна отображаться домашняя страница WordPress, внизу которой будет находиться информационный журнал Yii2. Благодаря моделям Yii2 можно выполнять различные команды DAO, renderPartials и тому подобные внутри функций WordPress, плагинов, шаблонов страниц и тому подобных.

```

1  <?php
2  class ExceptionHandler
3  {
4      public function __construct()
5      {
6          define('YII_ENABLE_EXCEPTION_HANDLER', false);
7          set_exception_handler(array($this, 'handleException'));
8      }
9
10     public function handleException($exception)
11     {
12         // отключение захвата ошибок, чтобы избежать рекурсивных ошибок
13         restore_error_handler();
14         restore_exception_handler();
15
16         $event=new CExceptionEvent($this,$exception);
17         if($exception instanceof CHttpException && $exception->statusCode == 404)
18         {
19             try
20             {
21                 Yii::app()->runController("wp/index");
22             }
23             catch(Exception $e) {}
24             // если создать исключение в Wordpress на 404, то можно использовать
25             // основной обработчик ошибок
26         }
27
28         if(!$event->handled)
29         {
30             Yii::app()->handleException($exception);
31         }
32     }
33 }

```

Рис. 4. Создание обработчика исключений WordPress

```

1  <?php
2
3  class WpController extends Controller
4  {
5      public function actionIndex()
6      {
7          $this->layout = false; // отключение макета
8          try {
9              $this->render('index');
10             Yii::app()->end();
11         }
12
13
14         catch (Exception $e) {
15             throw $e;
16         }
17     }
18 }

```

Рис. 5. Контроллер WordPress

```

1  <?php
2  wp();
3  require_once( ABSPATH . WPINC . '/template-loader.php' );

```

Рис. 6. Создание входящей точки в WpController

В результате интеграции CMS WordPress и Yii2 был разработан сайт, который объединяет маршруты WordPress и Yii2 в рамках одного пути, что в свою очередь позволяет настраивать /widgets и /customers таким образом, чтобы они указывали на контроллеры Yii2, а /about-us, /contact-us, /blog/* и /faq на страницы WordPress.

Библиографический список

1. Романов Ю.Д., Копышева Т.Н. Интеграция модуля CMS Joomla с 1С: Предприятие для обработки информации интернет-магазина // В сборнике: Актуальные проблемы математических и технических наук. Чебоксары, 2018. С. 241-245.
2. Кузнецов А.А., Толстобров А.П. Разработка компонента для CMS Joomla! для интеграции платежной системы PayPal // В сборнике: Сборник студенческих научных работ факультета компьютерных наук ВГУ. Федеральное агентство по образованию Российской Федерации государственное образовательное учреждение высшего профессионального образования "Воронежский государственный университет" факультет компьютерных наук. Воронеж, 2009. С. 63-69.
3. Вегнер А.И. Исследование систем управления студенческими проектами и технологии интеграции внутренних и внешних систем // В сборнике: Прогрессивные технологии и экономика в машиностроении. Сборник трудов X Всероссийской научно-практической конференции для студентов и учащейся молодежи. Под редакцией Е.А. Зернина. 2019. С. 244-246.
4. Пургина М.В., Добрынин А.С., Койнов Р.С. К вопросу интеграции корпоративных web-порталов // Информационные ресурсы России. 2015. № 5 (147). С. 36-37.
5. Казакевич В.А., Бабешко В.Н. Использование фреймворков в современном программировании // В сборнике: Качество продукции: контроль, управление, повышение, планирование. Сборник научных трудов 8-й Международной молодежной научно-практической конференции. Курск, 2021. С. 147-149.
6. Эмомалиев М.Р., Попов А.А. Лучший фреймворк PHP для разработки высоконагруженных систем // Modern Science. 2019. № 7-2. С. 323-327.