

Міністерство освіти та науки України  
Тернопільський обласний комунальний  
інститут післядипломної педагогічної освіти

**СТВОРЕННЯ САЙТУ ЗА ДОПОМОГОЮ  
CMS MODX EVO**

Тернопіль — 2019

Рецензенти:

І. В. Гевко, доктор педагогічних наук, професор кафедри комп'ютерних технологій Тернопільського національного педагогічного університету імені Володимира Гнатюка;

С. В. Матринюк, кандидат фізико-математичних наук, доцент кафедри інформатики та методики її навчання Тернопільського національного педагогічного університету імені Володимира Гнатюка;

**Вельгач А. В.**

В27 Створення сайту за допомогою CMS ModX Evo. / уклад. А. В. Вельгач. — Тернопіль : ТОКІППО, 2019. — 48 с.

У посібнику представлено основні прийоми та методи розв'язання завдань, які ставлять перед розробником web-ресурсів при створенні web-сайтів за допомогою CMS ModX Evo.

Для учнів загальноосвітніх навчальних закладів в рамках вивчення вибіркового модуля «Веб-технології» Інформатики 10-11 клас, а також фахівцями з розробки веб-ресурсів для ЗНЗ.

удк 004.738.52

## ВСТАНОВЛЕННЯ ТА НАЛАШТУВАННЯ ЛОКАЛЬНОГО СЕРВЕРА ХАМПП

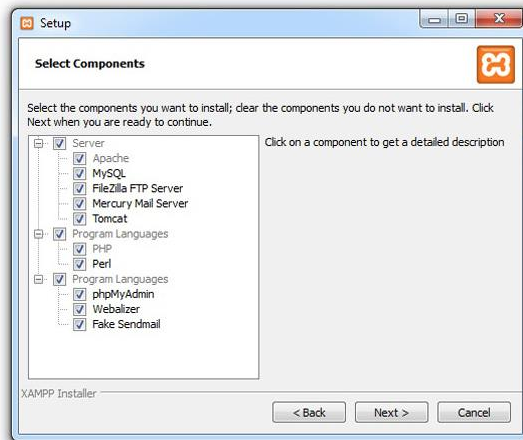
Локальний сервер — це набір програм, які дозволяють розробляти сайт на локальному комп'ютері без підключення до Інтернету. Він аналогічний до серверів, які розташовані у хостерів. Якщо створювати сайт, використовуючи тільки HTML-сторінки з дизайном CSS, то ніякого локального сервера використовувати не потрібно. Але якщо сайт динамічний, тобто створений з використанням Php, Perl, MySQL, скриптів тощо, то для його налагодження і тестування знадобиться web-сервер. Одним із таких дистрибутивів, який використовують web-розробники на локальному Windows-комп'ютері, є ХАМПП. Базовий пакет містить більшість необхідних програм і утиліт:

- інсталятор (підтримується також інсталяція на flash-накопичувач);
- Apache, SSL, SSI, mod\_rewrite, mod\_php;
- PHP5 з підтримкою GD, MySQL, sqLite;
- MySQL5 з підтримкою транзакцій;
- система управління запуском і завершенням всіх компонентів ХАМПП;
- phpMyAdmin — система управління MySQL через web-інтерфейс.

### *Встановлення ХАМПП*

Спочатку необхідно завантажити інсталятор ХАМПП на сторінці підтримки дистрибутива: <https://www.apachefriends.org/ru/download.html>. Після завантаження необхідно запустити інсталятор.

Після запуску з'явиться вікно, у якому користувачеві пропонується вибрати компоненти для встановлення.



**Рис. 1.** Вікно вибору компонентів ХАМПП

Практично всі зазначені компоненти можуть стати в нагоді в подальшому, тому краще залишити всі запропоновані опції за замовчуванням. Для базової роботи знадобляться Apache, MySQL, PHP. Також бажано встановити phpMyAdmin для більш комфортної роботи з базами даних.

Після натиснення **Next** у наступному вікні потрібно вибрати цільову папку встановлення. У наступному вікні буде запропоновано встановити Bitnami для XAMPP, тобто набір плагінів і дистрибутивів для різних CMS (наприклад, Drupal, Joomla, WordPress). Ця опція покликає спростити життя новачкам, але на практиці все виявляється значно важчим. Тому рекомендують зняти цей прапорець і натиснути **Next** для продовження встановлення. Після проходження кількох наступних кроків з'явиться вікно з повідомленням про успішне завершення процесу інсталяції XAMPP. Після натиснення кнопки **Finish** у цьому вікні буде здійснено перехід до панелі управління сервером.

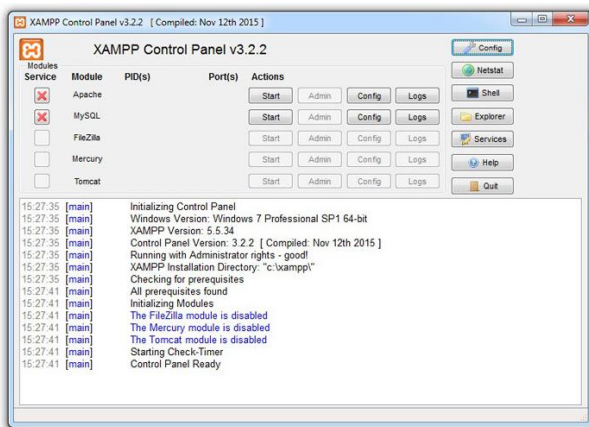


Рис. 2. Вікно панелі управління сервером XAMPP.

Для початку роботи з сервером досить натиснути кнопку **Start** навпроти сервера Apache і бази даних MySQL (рис. 2). Успішно запуснені служби будуть підсвічені зеленим фоном та їх кнопки **Admin** стануть активними.

Кнопка **Admin** Apache відкриває посилання <http://localhost/dashboard/> у браузері, де можна знайти набір корисних даних: питання і відповіді щодо XAMPP, документацію і запуск phpinfo (інформацію про поточні налаштування і конфігурацію PHP). Кнопка **Admin** MySQL запускає phpMyAdmin — інструмент для роботи з базою даних.

Для того щоб запустити файли сайту для тесту на локальному сервері XAMPP, достатньо скопіювати всі файли сайту в окремий каталог у каталозі сервера, обраному при встановленні (за замовчуванням, це каталог **htdocs**, повний шлях до нього `C:\xampp\htdocs\`). Якщо ж було змінено папку призначення, то `C:\вибрана_папка_при_встановленні\htdocs\`.

Наприклад, якщо каталог з файлами сайту має назву **sait**, то після копіювання на локальний сервер шлях до файлів сайту буде таким: **C:\xampp\htdocs\sait\**, а доступ до сайту в браузері — **http://localhost/sait/**. Увівши в браузері даний шлях, буде здійснено «запит» до файлів локального сервера до у каталозі **sait**, після чого відкриється індексний файл сайту (наприклад, *index.html*, або *index.php* тощо).

### **Створення віртуального хоста**

Створення віртуальних хостів — це хороший спосіб розташування кількох доменів за однією IP-адресою. Це дозволяє кільком сайтам користуватися загальними параметрами системи (оперативною пам'яттю тощо). Таку технологію часто використовують компанії, що надають віртуальний хостинг.

На локальному сервері можна налаштувати переадресацію будь-яких запитів браузера на певну папку локального web-сервера. Наприклад, при наборі в адресній стрічці браузера адреси **http://mysite/** буде здійснено запит до відповідного каталогу **C:\XAMPP\htdocs\mysite** сервера.

Для цього спочатку потрібно додати переадресацію сайтів у файл **hosts**, додавши в кінці файлу стрічку: **127.0.0.1 mysite**. Цей файл міститься у каталозі **etc** за адресою **C:\Windows\System32\drivers\etc**.

Після редагування файлу **hosts** необхідно здійснити налаштування самого сервера. Для цього потрібно відкрити, наприклад, у блокнуті файл **httpd-vhosts.conf**, який міститься у каталозі **C:\XAMPP\apache\conf** (це можна здійснити шляхом натиснення кнопки **Config** Apache у вікні керування сервером XAMPP) і в кінці файла дописати текст:

```
<VirtualHost mysite:80>  
    DocumentRoot "C:/XAMPP/htdocs/mysite"  
    ServerName mysite  
</VirtualHost>
```

Зберігши відредагований конфігураційний файл, необхідно перезавантажити Apache, щоб нові налаштування активізувалися. Тепер, якщо в каталог **C:\XAMPP\htdocs\mysite** зберегти файл **index.html** або **index.php** і в браузері перейти за адресою **http://mysite**, то відкриється головна сторінка віртуального домена.

### **Створення бази даних MySQL**

Використання бази даних MySQL є однією з умов функціонування CMS ModX Evo. Тому перш ніж приступити до встановлення системи керування контентом для майбутнього сайту, потрібно створити окрему базу даних. Для роботи з базою даних MySQL у XAMPP вбудовано web-додаток **phpMyAdmin**, який можна запустити шляхом натискання кнопки **Admin** MySQL або перейшовши за адресою **http://localhost/tools/phpMyAdmin**.



Рис. 3. Вікно web-додатку phpMyAdmin

Кожна база даних повинна мати свою унікальну назву. Придумавши таку назву, потрібно вибрати закладку **Бази даних**, внизу у вікні **Нова база даних** необхідно записати назву бази даних майбутнього сайту і в полі **Порівняння** обрати `utf8_general_ci` — це кодування для бази.

Після створення бази даних для неї потрібно створити користувача, якому необхідно призначити необхідні привілеї (у **Глобальних привілеях** потрібно обрати **Відмітити все** і натиснути **Ok**).

## CMS MODX EVOLUTION

MODX Evolution — система управління контентом сайту з відкритим вихідним кодом. Розповсюджується абсолютно безкоштовно. ModX Evolution використовують для сайтів практично будь-якої складності — чи це сайт-візитка на три сторінки, чи інтернет-магазин на кілька тисяч одиниць товару. Найбільшою перевагою цієї системи є те, що для web-сайту, створюваного за допомогою CMS ModX, можна використовувати звичайний HTML-CSS-шаблон. Адміністративна панель ModX Evo багатомовна з підтримкою української мови.

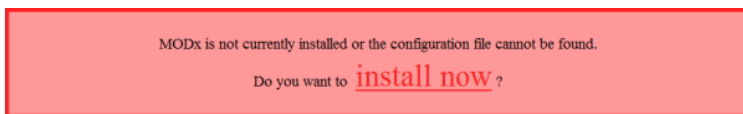
### *Встановлення MODX Evolution*

Для встановлення CMS ModX Evolution необхідно завантажити його дистрибутив необхідної версії за посиланням <http://modx.com.ua/download/> (версія 1.4.2 від 23.03.2018) і розпакувати отриманий архів у каталог `C:/XAMPP/htdocs/mysite`, після чого його вміст матиме вигляд (рис. 4):

Имя	Тип	Размер
assets	Папка с файлами	
install	Папка с файлами	
manager	Папка с файлами	
.gitignore	Файл "GITIGNORE"	1 КБ
ht.access	Файл "ACCESS"	2 КБ
index	Файл "PHP"	5 КБ
index-ajax	Файл "PHP"	1 КБ
sample-robots	Текстовый документ	1 КБ

**Рис. 4. Вигляд файлової системи CMS ModX Evo**

Після цього необхідно запустити Apache сервера XAMPP, якщо він не був запущений, і в браузері перейти за адресою: <http://mysite/>. У вікні браузера має з'явитися повідомлення-запрошення до інсталяції (рис. 5).



**Рис. 5. Вікно запрошення встановлення ModX Evo**

Перейшовши за посиланням **install now** і обравши мову інсталяції, переходимо до наступного кроку, де необхідно обрати пункт **Встановити нову копію Modx**, оскільки ModX встановлюється вперше.

Далі потрібно налаштувати з'єднання з базою даних. Для цього слід вказати ім'я хоста (localhost) та ім'я користувача і пароль, що збігаються з тими, які були вказані при створенні нової бази даних і користувача. Після введення усіх даних слід перейти за посиланням нижче для перевірки правильності введених даних. Якщо з'єднання з базою даних відбулося вдало, з'явиться запрошення для введення імені бази даних і префіксу для таблиць, а також метод зіставлення і кодування (рис. 6).

Параметры подключения и входа на сервер базы данных

Введите данные для входа в базу данных и затем проверьте их.

Хост базы данных:

Имя пользователя:

Пароль:

→ [Нажмите здесь для проверки соединения с вашим сервером базы данных и получения сопоставления](#)

Подключение: **успех** - сопоставление базы данных доступно

Введите имя базы данных, созданной для MODx. Если у вас еще нет базы данных, то программа установит вас. В зависимости от конфигурации MySQL или прав пользователя базы данных процесс может завершиться.

Имя базы данных:

Префикс таблиц:

Метод сопоставления:

Сопоставление:

→ [Нажмите здесь для создания базы данных или для проверки, что такая база существует](#)

**Рис. 6. Вікно налаштування з'єднання з базою даних**

Якщо все буде задано вірно, то після переходу за посиланням, розміщеним нижче, буде запропоновано ввести ім'я, e-mail та пароль адміністратора. На цьому кроці процесу встановлення буде запропоновано список компонентів, які будуть встановлені. Якщо на момент встановлення є невідомим призначення тих чи інших компонентів, тому слід залишити все за замовчуванням і натиснути на кнопку **Install**. Після того як з'явиться вікно з повідомленням про успішне встановлення, у ньому обов'язково необхідно встановити прапорець **Знищити каталог і файли програми встановлення**. На цьому процес встановлення завершений.

## Налаштування ModX Evolution

Усі налаштування й операції, які надалі проводитимуться із сайтом, здійснюватимуться через адміністративну панель, адреса доступу до якої <http://mysite/manager/>. Після введення імені та пароля адміністратора, які були вказані при встановленні ModX, буде здійснено перенаправлення на сторінку системної конфігурації, де можна задати основні налаштування системи. Більшість параметрів можна залишити за замовчуванням, але все-таки розглянемо найосновніші з них. Зауважимо, що біля кожного параметра присутній його короткий опис.

Параметр	Значення	Опис
Статус сайту (site_status)	<input checked="" type="radio"/> Online <input type="radio"/> Offline	
Заголовок сайту (site_name)	Українська глянцевий ім і Франка	Введіть заголовок вашого сайту.
Обраний адрес e-mail (site_email)	velgand@ukr.net	Укажіть адрес e-mail, який буде отримувати в листів підтвердження реєстрації в нові ID.
Перша сторінка (site_index)	1	Введіть ID ресурсу, який ви хочете використовувати як стартовий (домашню) сторінку. ПРИМІТКА: переконайтеся, що цей ID привласнений відповідному ресурсу, і що цей ресурс опублікований.
Сторінка помилки 404 (site_404)	25	Введіть ID ресурсу, який ви хочете використовувати як сторінку помилки 404. Ресурс не знайдено. ПРИМІТКА: переконайтеся, що цей ID привласнений відповідному ресурсу, і що цей ресурс опублікований.
Сторінка Доступ заборонено (site_access_denied)	1	Введіть ID ресурсу, який повинен бути увімкненою, коли посилається сайт на заборонену сторінку (403 - доступ заборонено). ПРИМІТКА: переконайтеся, що цей ID привласнений відповідному ресурсу, і що цей ресурс опублікований.
Сторінка Сайт недоступний		

Рис. 7. Вікно системної конфігурації адміністративної панелі ModX Evo

### Вкладка «Сайт»

#### Заголовок сайту

Дозволяє задати назву сайту. У цьому полі можна ввести або одне слово, або назву домену, або навіть назву сайту з невеликим його описом.

#### Перша сторінка

Дозволяє задати ID сторінки, яка повинна бути стартовою (домашньою), на неї відвідувач потраплятиме при завантаженні сайту. За замовчуванням це ресурс із ID=1.



### **Сторінка помилки «404»**

У цьому полі необхідно ввести ID сторінки, на яку повинен потрапляти відвідувач, якщо у браузері набрано адресу неіснуючої сторінки або якщо було натиснено «бите» посилання на сайті. Таке може періодично траплятися, тому тут надається можливість направити користувача на спеціальну сторінку, що сповіщає про відсутність сторінки. За замовчуванням ID=1, це означає, що осіб, які потрапляють на неіснуючу сторінку, буде перенаправлено на головну сторінку сайту.

### **Сторінка «Доступ заборонений»**

Необхідно задати ID ресурсу, який бачитимуть відвідувачі, якщо спробують зайти на закриту сторінку, тобто сторінку, яку можуть відвідувати лише авторизовані користувачі (403 — доступ заборонений).

### **Статус сайту**

Можливо обрати один з двох варіантів: *онлайн* або *оффлайн*. Якщо обрати *оффлайн*, відвідувачі сайту побачать повідомлення про недоступність сайту й не зможуть переглянути сам сайт. Це може бути корисно при проведенні технічних робіт на сайті, відновленні дизайну, структури тощо, щоб обмежити доступ до сайту й залишити його тільки адміністраторові.

### **Сторінка «Сайт-Недоступний»**

Якщо з якихось причин статус сайту виставлено в *оффлайн*, то у цьому полі можна задати ID сторінки, яку повинні побачити відвідувачі, якщо спробують зайти на сайт. За замовчуванням сторінка не створена і відвідувач просто побачить повідомлення про недоступний сайт, яке можна змінити в полі

### **Повідомлення про недоступність сайту.**

#### **Шаблон за замовчуванням**

Потрібно вибрати шаблон, який буде застосовуватися до нових ресурсів за замовчуванням. При редагуванні ресурсу можна встановити будь-який шаблон. За замовчуванням встановлено шаблон **Minimal Template**.

#### **«Публікувати» за замовчуванням**

Якщо обрано значення **Так**, то нові сторінки після збереження відразу будуть публікуватися.

#### **«Кешований» за замовчуванням**

Якщо обрано **Так**, то нові ресурси після створення будуть кешуватися<sup>1</sup> за замовчуванням.

---

<sup>1</sup> Кешування інтернет-сторінок — це процес збереження часто запитуваних документів на (проміжних) проксі-серверах або комп'ютері користувача з метою запобігання їх постійного завантаження із сервера-джерела й зменшення трафіку. Як один із варіантів, кешування web-сторінок може здійснюватися за допомогою CMS конкретного сайту для зниження навантаження на сервер при великій відвідуваності. Це означає, що при повторному завантаженні користувачем однієї й тієї самої сторінки вона частково буде взята з кеша і завантаження займе менше часу.

### «Доступний для пошуку» за замовчуванням

У більшості випадків необхідно буде створювати сторінки, доступні для пошуку по сайту, тому слід залишити значення **Так**. За допомогою цього параметра можна сховати від внутрішньої пошукової машини на сайті різні службові сторінки.

### Різниця в часі

У цьому полі існує можливість обрати похибку (кількість годин) між часом на місці перебування й на місці знаходження сервера. Цей параметр можна буде виставити після публікації сайту в мережі Інтернет.

### Тип сервера

За замовчуванням встановлено **http**. За потреби можна встановити **https<sup>2</sup>**-з'єднання.

## Вкладка «Дружні URL»

### Використовувати дружні URL

Даний параметр дозволяє використання дружніх URL на сайті (так званих ЧПУ). Для цього слід встановити для параметра значення **Так**.

### Посилання у форматі XHTML

Якщо ввімкнений цей параметр, то всі амперсанди (&) у посиланнях будуть замінені на конструкцію &amp; відповідно до формату XHTML.

### Префікс для дружніх URL

Це поле слід залишити порожнім.

### Суфікс для дружніх URL

У цьому полі можна вказати суфікс для дружніх URL, який буде відділятися крапкою від псевдоніма сторінки. Зокрема, це матиме вигляд як розширення файлів, наприклад, .html, .htm, .php і інші. За замовчуванням встановлено .html, але за потреби його можна змінити або взагалі залишити порожнім.

### Використовувати псевдоніми в URL

Якщо значення у даному полі залишити **Так**, то в якості псевдоніма сторінки використовуватиметься значення, яке введене в поле **Псевдонім** ресурсу. Наприклад, якщо у цьому полі буде вписано about і при цьому буде залишено суфікс для дружніх URL як .html, тоді URL цієї сторінки матиме вигляд **http://ім'я\_сайту/about.html**. Якщо поле **Псевдонім** залишити порожнім, то URL згенерується, виходячи з ID ресурсу й матиме вигляд **http://ім'я\_сайту/2.html**, де 2 — ідентифікатор ресурсу.

---

<sup>2</sup> https (Hypertext Transfer Protocol Secure) — розширення протоколу http, що підтримує шифрування. Забезпечує захист переданих даних і використовується для того, щоб забезпечити аутентифікацію й захищене з'єднання. Широко використовується, наприклад, у платіжних системах в інтернет-магазинах для шифрування переданих даних про кредитні картки.

## Використовувати вкладені URL

Якщо у даному полі залишити значення **Так**, то на сайті одержимо вкладену структуру URL. Наприклад, якщо дочірній ресурс має псевдонім **дочірній**, а батьківський ресурс — **батьківський**, то для дочірнього ресурсу отримаємо URL вигляду:

**http://ім'я\_сайту/батьківський/дочірній.**

## Автоматично генерувати псевдонім

Якщо в даному полі залишити **Так**, то система автоматично генеруватиме псевдоніми на основі заголовків сторінок. Тобто, якщо залишити при редагуванні або створенні ресурсу поле **Псевдонім** порожнім, у ньому значення згенерується автоматично на основі заголовку ресурсу. Якщо сайт англomовний і заголовки сторінок задані латиницею, то для включення автоматичної генерації псевдонімів досить лише цього перемикача. Якщо ж сайт україномовний і заголовки сторінок записано кирилицею, то необхідно додатково налаштувати плагін Transalias (після збереження налаштувань конфігурації). Для налаштування плагіна необхідно перейти до вкладки **Елементи** → **Керування елементами** → **Плагіни** → **Transalias** → **Вкладка конфігурація** → **Параметр Trans table виставити в russian** і зберегти.

## Вкладки «Користувачі» і «Безпека»

У вкладці «Користувачі» можна редагувати шаблони листів, що відправляються користувачам та менеджерам при реєстрації, зміні пароля тощо. У вкладці «Безпека» можна налаштувати кількість невдалих спроб входу в систему, які дозволені перш ніж користувач буде заблокований (щоб зловмисники не намагалися дібрати пароль). Для посилення безпеки при авторизації можна включити захисний код. При цьому для того щоб увійти в адміністративну панель, необхідно буде ввести код, нерозпізнаваний різного роду скриптами (показаний у вигляді графічного зображення). У полі «Слова для генерації Captcha-Кодів» можна ввести слова, які використовуватимуться для генерації цих захисних графічних зображень.

## Вкладка «Інтерфейс і представлення»

У цій вкладці можна здійснити налаштування, які мають стосунок до панелі адміністратора. Зокрема, з випадаючого списку можна обрати мову, на яку буде повністю переведена адміністративна частина системи керування. Це доволі таки зручно з ряду причин. По-перше, можна працювати при певних мовних навичках із клієнтами з різних країн, при цьому надавши їм можливість керувати своїм сайтом рідною мовою. Список підтримуваних мов доволі великий. По-друге, не знадобиться шукати локалізації системи на потрібну мову. По-третє, переклад, наданий системою, є професійним повним перекладом, тобто перекладені абсолютно всі сторінки й другорядні елементи (спливаючі підказки, коментарі тощо).

Також існує можливість встановити кодування сторінок сайту. Зі списку, що випадає, необхідно обрати «Unicode (UTF-8) — utf-8».

### Вкладка «Файл-Менеджер»

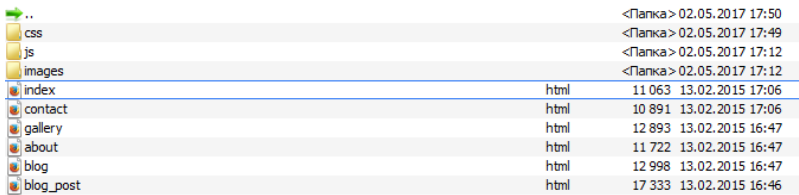
Тут через кому можна вказати, файли яких розширень можуть бути завантажені на сервер через файл-менеджер, їх максимально допустимий розмір і права для новостворених каталогів і завантажених файлів.

## ІНТЕГРАЦІЯ ШАБЛОНУ В СИСТЕМУ УПРАВЛІННЯ

Після встановлення і налаштування системи управління контентом необхідно інтегрувати шаблон з обраним дизайном для майбутнього сайту. Особливістю CMS ModX є те, що в якості шаблону підійде довільний темплейт HTML+CSS. Як приклад, розглянемо інтеграцію шаблону від [templatemo.com](http://templatemo.com). В обраному шаблоні організовані сторінки різного вигляду:

- головна сторінка з jQuery-слайдером;
- типова текстова сторінка (about);
- сторінка для перегляду переліку новин і їх детального перегляду (blog і blog\_post);
- сторінка фотогалереї (gallery);
- сторінка з формою зворотного зв'язку (contacts).

У завантаженому архіві міститься 3 каталоги і кілька HTML-файлів.



..		<Панка> 02.05.2017 17:50
css		<Панка> 02.05.2017 17:49
js		<Панка> 02.05.2017 17:12
images		<Панка> 02.05.2017 17:12
index	html	11 063 13.02.2015 17:06
contact	html	10 891 13.02.2015 17:06
gallery	html	12 893 13.02.2015 16:47
about	html	11 722 13.02.2015 16:47
blog	html	12 998 13.02.2015 16:47
blog_post	html	17 333 13.02.2015 16:46

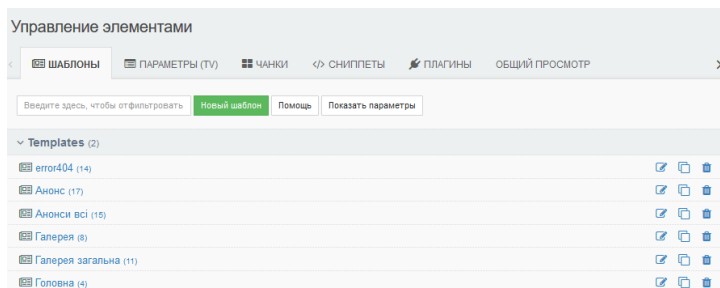
Рис. 8. Вигляд каталога завантаженого HTML+CSS темлейта

У каталозі **images** зберігаються зображення, які використовують для дизайну сайту, в каталозі **js** — скрипти, що підключаються для роботи jQuery слайдером на головній сторінці та детального перегляду фото на сторінці галереї, а в каталозі **css** зберігаються CSS-файли, які визначають зовнішній вигляд обраного шаблону.

HTML-файли, які містяться в архіві, є HTML-розміткою для кожної зі сторінок: `index.html` — головна сторінка, `contact.html` — сторінка з контактною інформацією і формою зворотного зв'язку, `gallery.html` — сторінка з галереєю зображень, `blog.html` і `blog_post.html` — сторінки з переліком постів і детальним переглядом поста відповідно, `about.html` — звичайна сторінка з текстовою інформацією. Після детального аналізу та вивчення всіх сторінок можна почати інтеграцію дизайну в систему управління.

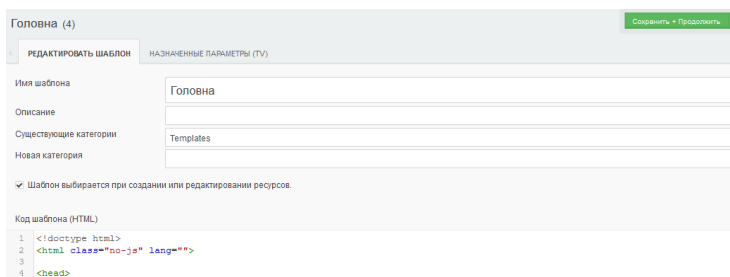
Усі файли темплейту, які використовуватимуться шаблонами сайту, доцільно зберігати в каталозі **assets/templates**, у який слід розпакувати вміст завантаженого архіву. Після того як усі файли опиняться в призначеному для них місці, потрібно авторизуватися в системі управління контентом (<http://mysite/manager>) для подальшого інтегрування шаблонів дизайну.

В адміністративній панелі переходимо до вкладки **Елементи** → **Шаблони**, у якій створюються нові або налаштовуються вже створені шаблони. Для знищення або редагування необхідно натиснути на посилання із назвою шаблону.



**Рис. 9.** Вікно керування шаблонами CMS ModX Evo

Для того щоб створити шаблон для головної сторінки сайту зі слайдером, слід натиснути на кнопку **Новий шаблон** і заповнити елементи форми, що з'явилася. Для цього в описі шаблону в полі **Ім'я шаблону** потрібно задати назву шаблону, а в полі **Опис** — короткий опис, який слід задавати, щоб не заплутатися, якщо шаблонів буде дуже багато. У поле **Код шаблону (HTML)** потрібно вставити HTML-розмітку головної сторінки, яка вказана у файлі `index.html`. Після заповнення усіх необхідних полів слід натиснути на кнопку **Зберегти** у верхній частині сторінки.



**Рис. 10.** Форма редагування опису шаблону

Для того щоб переглянути створений шаблон, його спочатку необхідно підключити до ресурсу, який завантажуватиметься при початковому завантаженні сайту. Для цього необхідно обрати створений шаблон у полі **Шаблон** форми редагування опису ресурсу з ID=1 і перейти за посиланням

<http://mysite/>. Після перегляду очевидним є те, що вигляд сторінки далекий від оригіналу, оскільки порушилися всі шляхи до файлів стилів, зображень і скриптів, тому їх необхідно змінити. Для цього слід перейти на сторінку редагування шаблону: **Елементи** → **Шаблони** → **Головна**. Оскільки вкладеність каталогів змінилася при інтеграції дизайну в систему управління, то для всіх шляхів, які використовуються в HTML-коді (**css/**, **images/**, **js/**) необхідно додати таку конструкцію: **/assets/templates/**.

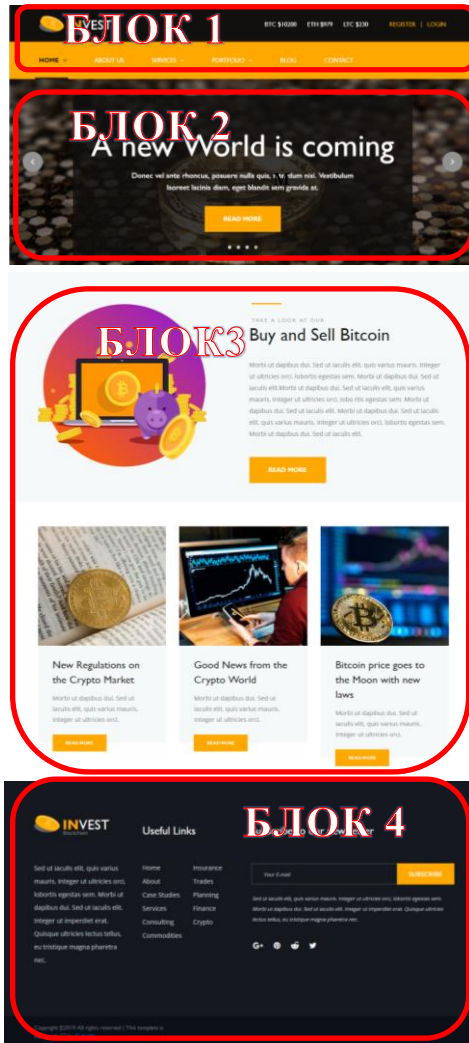


Рис. 11. Вигляд шаблону головної сторінки

Для прикладу, ось такий вигляд мав шлях підключення CSS-стилів в оригіналі: `<link href="css/style.css" rel="stylesheet" type="text/css" />`, а вигляд цього тегу після зміни шляху повинен бути таким:

```
<link href="/assets/templates/css/style.css" rel="stylesheet" type="text/css" />
```

Після зміни всіх шляхів необхідно зберегти змінений шаблон і знову переглянути головну сторінку сайту (**Сайт** → **Перегляд**). Якщо все відредаговано вірно, вигляд головної сторінки має стати ідентичним до головної сторінки завантаженого шаблону.

### *Налаштування шаблону, розбиття на чанки*

Для раціонального налаштування інтегрованого шаблону і швидкої зміни всього дизайну створюваного сайту даний шаблон слід розбити на логічні складові. Для цього слід проаналізувати, з яких блоків складається шаблон головної сторінки та шаблони решти сторінок, які планується інтегрувати в систему. Зокрема, в шаблоні головної сторінки як правило можна виділити такі блоки:

- **«шапка» сайту** (header, рис. 11, блок 1), у якій, як правило, розміщують логотип і головне меню навігації;
- **jQuery слайдер** (рис. 11, блок 2);
- **блок з анонсами або коротким змістом статей, новин тощо** (рис. 11, блок 3);
- **«підвал» сайту** (footer, рис. 11, блок 4), у якому розміщений вміст, який знаходиться внизу сторінки, а саме блок з інформацією із соціальних мереж, пости ззовні, блок із зазначенням авторських прав тощо.

Перш ніж розпочати на рівні розмітки розбивати код шаблону на складові частини, слід ознайомитися із терміном «чанк». Чанк — це не що інше як частина HTML-коду. Наприклад, є шаблон, у якому деяка частина HTML-коду повторюється більше одного разу, але не раціонально вносити зміни відразу скрізь, де зустрічається цей повторюваний код. Для цього в системі управління створюється чанк з ім'ям, наприклад **Чанк**, а в якості вмісту чанка використовується ця частина повторюваного коду. Після чого в розмітці шаблону замість цих повторюваних блоків вставляється конструкція `{ {Чанк} }`, яку ModX при генеруванні сторінки автоматично замінить на вміст цього чанка. Таким чином, при необхідності внести якісь зміни в повторювані блоки достатньо просто відредагувати створений чанк. Слід звернути увагу на те, що виклик чанка здійснюється за його іменем, записаним у подвійних фігурних дужках.

Може виникнути логічне запитання: чи потрібно ділити шаблон сторінки на чанки, якщо в ній немає блоків, що повторюються. Щоб отримати відповідь на це запитання, слід проаналізувати відразу всі шаблони сторінок сайту, що створюється. Як правило, можна помітити, що всі вони мають деякі однакові елементи (наприклад: «шапку» і «підвал» сайту), які логічно буде ви-

нести в окремі чанки, щоб мати можливість вносити зміни в усі шаблони одразу з одного місця, а саме при редагуванні чанку. Неможливо не погодитися з тим, що не дуже зручно при зміні інформації, розташованої в футері сайту або при внесенні змін до шапки сайту, редагувати кожен шаблон окремо. А якщо таких шаблонів буде дуже багато, то при цьому, крім витраченого часу, збільшується ймовірність того, що в процесі редагування будуть допущені механічні помилки.

Тому переходимо до вкладки управління шаблонами і відредагуємо створений шаблон головної сторінки **Головна**. Спочатку рекомендується винести в окремий чанк тег `<head>` зі всіма службовими тегами всередині нього і підключеними скриптами та стилями. Для цього знаходимо в HTML-коді шаблону цей тег і копіюємо його в буфер обміну. Після цього слідуємо за вказівкою: **Елементи** → **Чанки** і переходимо за посиланням **Новий чанк**. Для того щоб створити чанк для тегу `<head>`.

У формі, що з'явилася, заповнюємо поля **Назва чанка** (наприклад HEAD), вписуємо його опис, щоб легше було орієнтуватися, коли чанків буде багато, та обов'язково в поле **Код чанка** вставляємо вміст буферу обміну. Також можна вписати назву категорії, в яку буде поміщено цей чанк. Це не обов'язково, але теж допоможе не заплутатися у великій кількості чанків. Назвемо категорію, наприклад, **Загальний дизайн сайту**.

head (3) Сохранить + Продолжить Сделать копию Удалить Отмена

ОБЩИЕ

Название чанка head

Описание

Существующие категории Templates

Новая категория

Отключен

Код чанка (HTML) Редактор: Нет

```
1 <meta http-equiv="Content-Type" content="text/html; charset=[(modx_charset)]" />
2 <title>[*title]</title>
3 <![noIndex]>
4 <meta name="keywords" content="[*key*]" />
5 <meta name="description" content="[*desc*]" />
6 <meta name="viewport" content="width=device-width, initial-scale=1">
7 <!-- Favicon -->
8 <link rel="shortcut icon" type="image/x-icon" href="/assets/templates/img/favicon.ico">
9 <!-- Normalize CSS -->
10 <link rel="stylesheet" href="/assets/templates/css/normalize.css">
11 <!-- Main CSS -->
12 <link rel="stylesheet" href="/assets/templates/css/main.css">
13 <!-- Bootstrap CSS -->
14 <link rel="stylesheet" href="/assets/templates/css/bootstrap.min.css">
15 <!-- Animate CSS -->
16 <link rel="stylesheet" href="/assets/templates/css/animate.min.css">
17 <!-- Font-awesome CSS-->
18 <link rel="stylesheet" href="/assets/templates/css/font-awesome.min.css">
```

Рис. 12. Форма створення чанка

Після збереження новоствореного чанка потрібно повернутися до редагування шаблону: **Елементи** → **Шаблони** → **Головна**, і замінити частину коду, яка була винесена в чанк на виклик чанка. Тобто замість тега `<head>` і всього, що в ньому було розташоване, необхідно вставити конструкцію `{ {HEAD} }`.



Аналогічним чином потрібно створити й інші чанки. Для цього необхідно вирізати потрібну частину коду з шаблону, вставити його вміст у чанк, а в шаблоні на його місці викликати створений чанк.

Рекомендується винести частини HTML-коду в такі чанки:

```
{{HEADER}}, {{FOOTER}}
```

У чанку `{{HEADER}}` рекомендується зберегти частину HTML-коду, що відповідає за виведення шапки сайту з логотипом і меню навігації. У чанк `{{FOOTER}}` поміщаємо контейнер, що відповідає за відображення нижньої частини сайту.

*Зауваження.* Зовнішній вигляд шаблону в браузері не повинен змінитися після того, як повністю виконано розбиття його на чанки.

## ДЕРЕВО РЕСУРСІВ І СТВОРЕННЯ СТОРІНОК

Для створення кількох сторінок на сайті розглянемо дерево ресурсів ModX, яке розміщується в лівій частині адміністративної панелі.

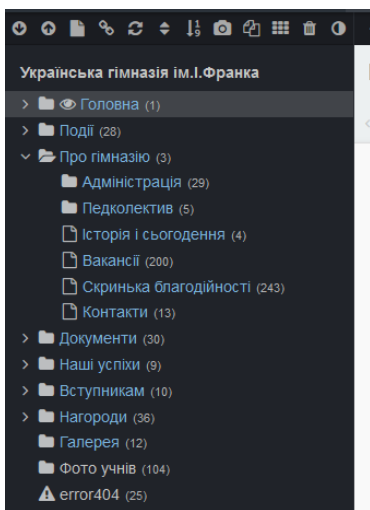


Рис. 13. Вигляд дерева ресурсів ModX Evo

Дерево ресурсів є структурою сайту. З його допомогою створюють і редагують усі сторінки та категорії на сайті. До речі, сторінки та категорії у ModX прийнято називати *ресурсами*. Деревоподібна структура дуже зручна для адміністрування сайтів і за рахунок своєї інтуїтивності буде зрозуміла навіть недовідченному користувачеві.

У корені дерева вказано назву сайту, яка задається на сторінці системної конфігурації у полі **Назва сайту**. За замовчуванням при інсталяції створено один ресурс із назвою **Home**, який і слугує головною сторінкою сайту. У дуж-

ках після назви вказано ID ресурсу — це ціле число, яке є унікальним для кожної сторінки, тобто не може бути двох різних ресурсів з одним і тим самим ідентифікатором. ID надалі будуть використовуватися для звернення до ресурсів.

Для редагування існуючого ресурсу слід натиснути по його назві в дереві ресурсів, після чого з'явиться форма редагування ресурсу.

Рис. 14. Форма редагування ресурсу. Вкладка «Загальні»

### Вкладка «Загальні»

Тут можна задати наступні атрибути ресурсу, а саме такі поля:

«**Заголовок**» — назва сторінки.

«**Розширений заголовок**» — у це поле вписують розширений заголовок сторінки. Можна це поле залишити порожнім або скопіювати сюди вміст поля **Заголовок**.

«**Псевдонім**» — тут потрібно вписати ім'я сторінки (латинськими літерами), яка буде відображатися в URL адресної стрічки браузера при відображенні сторінки. Якщо його залишити порожнім, воно заповниться автоматично, виходячи із заголовка сторінки, якщо налаштований плагін TransAlias. Якщо плагін не налаштований, то пусте поле буде змінено на ідентифікатор ресурсу.

«**Анотація**» — поле з коротким описом сторінки.

«**Шаблон**» — представляється у вигляді випадуючого списку. У цьому списку відображаються всі створені шаблони, які можна обирати для ресурсу (сторінки). Кількість шаблонів в ModX не обмежена.

«**Пункт меню**» — необов'язкове поле, яке, як правило, використовують, якщо заголовок сторінки дуже довгий і його незручно використовувати в якості назви пункту меню. У цьому випадку в якості назви пункту меню буде використовуватися це поле.

«**Позиція в меню**» — це порядковий номер ресурсу в меню, за допомогою цього параметра можна по-різному відсортувати відображувані в меню пункти.

«**Показувати в меню**» — прибравши даний прапорець з цього пункту, забороняється показ назви ресурсу в меню.

«**Папка**» — вказується, який батьківський документ буде в ресурсу. Для того щоб змінити батьківський документ, необхідно клікнути мишкою по іконці поруч, після чого обрати батьківський документ у дереві сайту.

На цій же закладці міститься і візуальний редактор, за допомогою якого можна редагувати весь вміст сторінок.

### **Вкладка «Налаштування сторінки»**

Тут вказують додаткові параметри для створюваного чи редагованого ресурсу.

Головна (1)

< ОБЩИЕ НАСТРОЙКА СТРАНИЦЫ SEO

Публиковать

Дата публикации  dd-mm-YYYY HH.MM.SS

Дата отмены публикации  dd-mm-YYYY HH.MM.SS

Тип ресурса

Тип содержимого

Местонахождение содержимого

Контейнер (содержит дочерние ресурсы)

Участвует в URL

Использовать HTML-редактор

Показывать дочерние ресурсы

Доступен для поиска

Кэшируемый

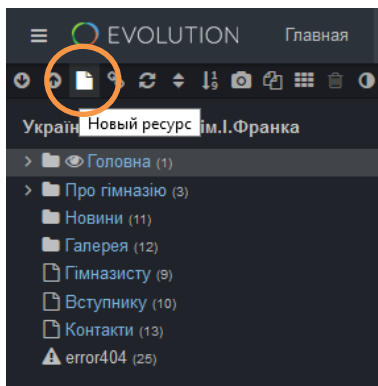
Очистить кэш

**Рис. 15. Форма редагування ресурсу. Вкладка «Налаштування сторінки»**

**Прапорець «Публікувати».** З його допомогою можна дозволяти або скасовувати публікацію того чи іншого ресурсу. Тут також можна встановлювати дату публікації та дату її відміни.

Інші параметри в більшості випадків можна буде залишати за замовчуванням. Прапорці, які будуть виставлятися при редагуванні ресурсу за замовчуванням, можна визначати на сторінці конфігурації ModX. Ці налаштування були описані вище при розгляді питання налаштування ModX.

Для створення нового ресурсу потрібно натиснути на відповідну піктограму, розташовану вище дерева документів.



**Рис. 16.** Створення нового ресурсу у дереві ресурсів

## РЕАЛІЗАЦІЯ ДИНАМІЧНОГО МЕНЮ

У CMS ModX є ряд вбудованих рішень для розв'язання окремих завдань, які можуть виникнути перед розробником сайту. Зокрема, однією із таких класичних задач є організація динамічного меню на сайті. Реалізувати це можливо за допомогою сніппета **DLMenu**. Термін «сніппет» — ще один термін, який необхідно знати для роботи з CMS ModX.

Сніппет — це PHP-код, який відповідно до запрограмованого алгоритму генерує HTML-код, що вставляється у шаблон сайту в місце виклику. Щоб переглянути список фрагментів, які йдуть в комплекті дистрибутива CMS ModX за замовчуванням, потрібно перейти у вкладку адміністративної панелі **Елементи** → **Сніппети**. Усі вони доступні для редагування. Для чого ж потрібні сніппети? Потрібні вони для забезпечення динамічної логіки на сайті. Наприклад, для генерації динамічного меню, виведення контенту з бази даних за якоїсь умови, для організації форми зворотного зв'язку, та й загалом, для реалізації стандартних задач, доступних через ModX API.

Виклик сніппета здійснюється за його іменем, що поміщається між квадратними дужками зі знаками оклику. Наприклад, якщо ім'я певного сніппета **Name**, то виклик його здійснюватиметься конструкцією, яка має бути розміщена у необхідному місці шаблону: `[!Name!]`.

Крім виклику сніппета, йому можна відправляти на обробку різні значення параметрів, якщо такі підтримуються даним сніппетом. Зазвичай, у кожного сніппета є свій набір підтримуваних параметрів, від яких залежатиме результат роботи сніппета. Для того щоб передати деякі значення параметрів, використовують конструкцію

```
[!Name? &параметр1 = `значення` & параметр2 = `значення`!]
```

Тут необхідно звернути увагу на знак запитання, який ставлять після імені сніппета. Він вказує на те, що далі вказується перелік параметрів, які передаються. Перед назвою кожного параметра, який передається, ставиться знак «&», а його значення вкрито в зворотні одинарні лапки.

Зауважимо, що вище описано виклик некешованого сніппета. Якщо результат роботи необхідно занести в кеш, то виклик здійснюватиметься за іменем сніппета, поміщеному в подвійні квадратні дужки. Тобто для сніппета з ім'ям **Name** кешований виклик буде представляти собою таку конструкцію:

```
[ [Name] ]
```

При кешованому виклику результат роботи сніппета заноситься в кеш, і при повторному завантаженні сторінки код сніппета не виконується заново, а в місце виклику одразу підставляється попередній результат, що береться із кеша сайту на web-сервері.

Повернімося до задачі побудови динамічного меню і сніппета **DI Menu**.

Результатом роботи **DI Menu** є HTML-код, який є нумерованим списком, елементами якого будуть посилання на дочірні документи для тієї сторінки, для якої викликається сніппет. Для прикладу, якщо у нас структура документів така, як зображено на рисунку нижче,

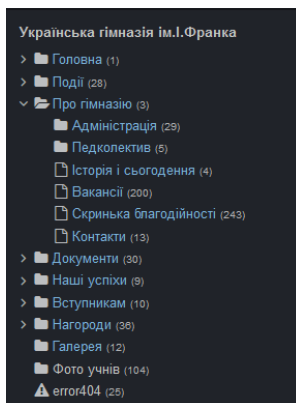


Рис. 17. Створення нового ресурсу у дереві ресурсів

то, результатом виклику сніппету `[!DI Menu!]` на сторінці «Про гімназію» буде список із 6 посилань на дочірні сторінки. HTML-код списку матиме вигляд:

```
<ul>
```

```
<li><a href="/about/history">Історія і сьогодення</a></li>
<li><a href="/about/administraciya/">Адміністрація</a></li>
<li><a href="/about/pedkolektiv/">Педколектив</a></li>
<li><a href="/about/vakansii">Вакансії</a></li>
<li><a href="/about/skrinka">Скринька благодійності</a></li>
<li><a href="/about/contacts">Контакти</a></li>
</ul>
```

При такому виклику обов'язковим є виконання умови наявності дочірніх елементів, адже якщо їх не буде, то сніппет виведе порожню стрічку. Такий виклик для організації меню на сайті не підходить, оскільки на всіх сторінках сайту має відображатися меню, що складається з усіх ресурсів дерева, які є не-прихованими для меню. Для того щоб це зробити, необхідно викликати сніппет **DLMenu** і передати йому параметр **parents**, у який слід передати значення ID того ресурсу, посилання на дочірні сторінки якого необхідно вивести. Тобто для побудови головного меню виклик матиме такий вигляд:

```
[!DLMenu?&parents = 0!]
```

У цьому випадку, на якій би сторінці не був викликаний сніппет, список буде формуватися з посилань на дочірні елементи головної сторінки (ID = 0), а не тієї, на якій було зроблено виклик.

Меню, зазвичай, є звичайним нумерованим списком. Це означає, що HTML-розмітка меню буде звичайним списком, а всю задачу за зовнішній вигляд бере на себе CSS-файл, який підключений у заголовку документу. Для виведення динамічного меню необхідно видалити з чанка **HEADER** код, що відповідає за виведення меню, а на його місці викликати сніппет **DLMenu** з параметром `parents='0'`:

```
[[DLMenu? &parents = 0]]
```

За замовчуванням, для активного пункту меню сніппет **DLMenu** завжди буде генерувати `class="active"`, тому, якщо в шаблоні він описується іншим класом, потрібно змінити його в CSS-файлі.

Якщо ж при виведенні в шаблоні меню (нумерованого списку) використовуються деякі класи, то їх можна описати при виклику сніппета через відповідні параметри, значеннями яких є відповідні шаблони. Список параметрів є досить таки великим і їх можна знайти в **додатку 2** або за посиланням: [http://docs.evo.im/04\\_extras/dlmenu.html](http://docs.evo.im/04_extras/dlmenu.html)

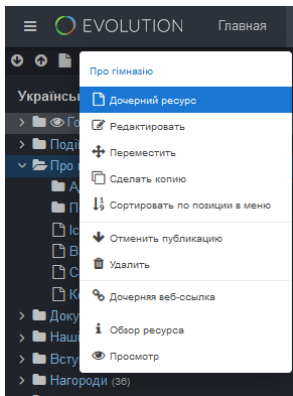
## СТВОРЕННЯ СТРІЧКИ НОВИН. СНІППЕТ DOCLISTER

Розглянемо питання про виведення стрічки новин на сторінках сайту. Для того щоб було що виводити, необхідно створити кілька дочірніх ресурсів із новинами, батьківським ресурсом для яких повинен бути ресурс із назвою **Новини**.

Дочірні ресурси можна створити двома способами (див. рис. 16):

1. Натисненням на піктограму **Новий ресурс** і у вкладці із загальними налаштуваннями ресурсу необхідно змінити значення параметра **Папка**, для якого вказати потрібний батьківський ресурс.

2. У дереві ресурсів будь-якою кнопкою миші клікнути на піктограму, розташовану близько назви ресурсу, для якого необхідно створити дочірню сторінку (клікаємо мишкою саме на піктограму, а не на текст), після чого з'явиться контекстне меню даного пункту. У контекстному меню необхідно обрати **Дочірній ресурс**.



**Рис. 18.** Контекстне меню у дереві ресурсів

Крім створення дочірнього ресурсу, у цьому контекстному меню доступно ще багато корисних функцій. Наприклад, можна перейти до редагування ресурсу, перемістити його в іншу папку в дереві ресурсів, зробити копію, опублікувати або скасувати публікацію, вилучити або відновити ресурс.

Після створення дочірніх ресурсів ресурс **Новини** повинен мати такий вигляд:

№	заголовок	ДАТА СТВОДАННЯ	ДАТА ПУБЛІКАЦІЇ	СТАТУС	ДЕЙСТВИЕ:
27	Тижень пам'яті Небесної Сотні	13-03-2019	22-02-2019	Опублікован	🔍 ⬆️ ⬇️ 🗑️
26	Флешмоб «Мій «Кобзар»	12-03-2019	11-03-2019	Опублікован	🔍 ⬆️ ⬇️ 🗑️
20	<u>Нагородження знавців української мови</u>	07-03-2019	21-02-2019	Опублікован	🔍 ⬆️ ⬇️ 🗑️
19	Жива картина	07-03-2019	01-03-2019	Опублікован	🔍 ⬆️ ⬇️ 🗑️
18	Розважальне ток-шоу «Валентинів день»	07-03-2019	14-02-2019	Опублікован	🔍 ⬆️ ⬇️ 🗑️

**Рис. 19.** Вигляд ресурсу «Новини» з дочірніми ресурсами

Для виведення стрічки новин на сторінці будемо використовувати вбудований сніппет **DocLister**. Це дуже потужний інструмент, який, напевне, є одним із основних вбудованих інструментів у CMS ModX Evolution. За допомогою **DocLister** можна на сайті виводити ресурси, які створені в дереві ресурсів, абсолютно в будь-якому вигляді. Така гнучкість досягається за рахунок роботи із шаблонами виведення.

За допомогою цього сніппета можна реалізувати: виведення статей і новин на сайті, виведення заміток у блозі, архів статей, RSS-стрічки, виведення даних у XML-форматі та ще багато корисного. Одним словом, це дуже функціональний сніппет, який будемо використовувати ще не раз.

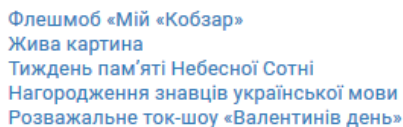
**DocLister** має дуже великий набір параметрів, передаючи які під час виклику сніппета, можна змінювати результат його роботи. Зі списком параметрів і можливими їх значеннями можна ознайомитися у **Додатку 3** або за посиланням [http://docs.evo.im/04\\_extras/doclistner.html](http://docs.evo.im/04_extras/doclistner.html).

Отже, виклик сніппета **DocLister** здійснюється за аналогічними правилами виклику сніппета **DIMenu**:

```
[ [DocLister? &параметр1=`значення` &параметр2=`значення` ] ]
```

Викликати сніппет необхідно безпосередньо у шаблоні, закріпленому за ресурсом із назвою **Новини** в області вмісту сторінки.

Якщо викликати **DocLister** без переданих параметрів, то результатом роботи цього сніппета буде виведення дочірніх ресурсів у такому вигляді:



Флешмоб «Мій «Кобзар»  
Жива картина  
Тиждень пам'яті Небесної Сотні  
Нагородження знавців української мови  
Розважальне ток-шоу «Валентинів день»

**Рис. 20. Вигляд виведення дочірніх ресурсів сніппетом DocLister без параметрів**

Як видно, на сторінці було виведено перелік усіх дочірніх ресурсів для сторінки **Новини**. При виведенні використовувалося поле **Заголовок**, яке є посиланням на сторінку дочірнього ресурсу, тобто з конкретною новиною. Для того щоб змінити зовнішній вигляд стрічки новин, необхідно здійснити виклик сніппета **DocLister** з деякими параметрами.

Насамперед, потрібно навчитися працювати із шаблонами виведення інформації, використання яких дозволить змінювати зовнішній вигляд новинної стрічки. У якості шаблону **DocLister** може використовуватися створений чанк або існує можливість безпосередньо задавати цей шаблон при виклику сніппета.

Для того щоб створити шаблон у чанку, необхідно перейти на сторінку із чанками **Елементи** → **Чанки** і створити новий чанк із будь-яким інтуїтивно



зрозумілим іменем, наприклад, **news-article**. Надалі він буде слугувати шаблоном виведення кожної новини на сайті. У цей шаблон помістимо назву, короткий зміст новини й дату публікації, при цьому заголовок має бути посиланням на сторінку ресурсу з новиною.

Слід звернути увагу на те, що в коді шаблону будуть використовуватися як спеціальні теги ресурсів ModX, що описані нижче, так і спеціальні теги **DocLister**.

Для розв'язання поставленого завдання знадобляться такі теги (плейсхолдери):

- **[+longtitle+]** або **[+pagetitle+]** — для виведення заголовку новини
- **[+date+]** — для виведення дати публікації
- **[+introtext+]** — для виведення короткого змісту новини

У щойно створений чанк **news-article** поміщаємо згадані вище теги, поки що без HTML-розмітки:

```
[+date+] [+pagetitle+] [+introtext+]
```

У шаблоні ресурсу **Новини** викликаємо сніпсет **DocLister** із вказівкою назви чанка, який буде використовуватися у якості шаблону. При цьому конструкція виклику сніпсета набуде такого вигляду:

```
[[DocLister? &tpl=`news-article`]]
```

Після цього при перегляді сторінки **Новини** стрічка новин буде стрічкою, виведеною без пробілів і без розмітки, точно так само, як зазначено в шаблоні створеного чанка **news-article**.

Тому для того щоб організувати виведення стрічки новин згідно шаблону, необхідно скопіювати частину HTML-коду, яка відповідає за виведення однієї новини, замінивши текст у ньому на відповідні теги (плейсхолдери).

Для прикладу, вміст чанка після редагування може набути такого вигляду:

```
<div id="news">
<h2><a href="[~[+id+]~]" title="[+pagetitle+]">[+pagetitle+]</a></h2>
<div class="introtext"><span class="date">[+date+]</span> [+introtext+]</div>
<div class="readmore" ><a href="[~[+id+]~]" title="Перейти до повного тексту новини">Читати
далі →</a></div>
</div>
```

де конструкція **[~[+id+]~]** — це URL ресурсу з ідентифікатором **id**.

При цьому дата, зазначена в стрічці новин, — це дата створення ресурсу, яку можна змінити шляхом використання параметра **&datesource = `pub\_date`**:

```
[[DocLister? &tpl=`news-article` &datesource = `pub_date` ]]
```

Таким чином, у якості джерела дати публікації буде використана дата, розміщена у вкладці **Налаштування сторінки**. Це поле можна змінити для кожного ресурсу.

Для того щоб змінити формат виведення дати (наприклад до вигляду ДД.ММ.РРРР), необхідно використати ще один параметр **DocLister &dateformat**. Конструкція виклику сніпета матиме такий вигляд:

```
[[DocLister? &tpl=`news-article` &datesource = `pub_date`  
&dateformat=`%d.%m.%Y`]]
```

Якщо на сторінці перелік введених дочірніх ресурсів є доволі великим, то таке виведення слід виконувати посторінково. Тобто стрічку новин необхідно розбивати на окремі сторінки та забезпечити перехід від сторінки до сторінки за допомогою пейджингу. Для розбивки на сторінки новин необхідно в конструкцію виклику сніпета додати такі параметри:

**&paginate** — параметр, який вказує на те, чи включити або чи відключити посторінкову розбивку. Щоб включити розбивку, необхідно вказати значення **pages**;

**&display** — параметр, який вказує на кількість новин, що необхідно вивести на одній сторінці. Оскільки зараз створено 5 новин, то, щоб продемонструвати розбивку, необхідно виставити значення менше 5, наприклад, 2 ресурси.

Після додавання зазначених параметрів і плейсхолдерів виклик сніпета, який розташований у шаблоні ресурсу **Новини**, буде мати таку конструкцію:

```
[[DocLister? &tpl=`news-article` &datesource = `pub_date`  
&dateformat=`%d.% m.% Y` &id=`news` &paginate=`pages`  
&display=`2` ]]  
Сторінки: [+news.pages+]
```

Необхідно зауважити, що значення в параметрі **id** і плейсхолдері виклику нумерації сторінок повинні бути ідентичними (у наведеному прикладі це **news**). Результатом цього виклику повинна стати розбита на сторінки стрічка новин.

## TV-ПАРАМЕТРИ MODX. ДОДАВАННЯ ЗОБРАЖЕНЬ ДО РЕСУРСІВ.

Для кожного шаблону в CMS ModX є можливість додати додаткові поля, які згодом можна виводити на сторінці за допомогою спеціальних плейсхолдерів. Це дає небувалу гнучкість у розробці. Використовувати TV-параметри будемо на прикладі додавання зображень до новин і виведення цих зображень у стрічці новин з коротким описом новини.

## Створення TV-параметра


Для створення TV-параметра слід перейти до вкладки **Елементи** → **TV параметри**, в якій натиснути на посилання **Новий параметр (TV)**.

У формі створення TV-параметра слід заповнити такі поля:

- **Ім'я параметра.** Уводимо назву параметра, за якою він буде викликатися в шаблоні. Бажано надати осмислені назви, щоб було легше зорієнтуватися при редагуванні шаблонів. Для додавання зображень до новин можна обрати для прикладу назву **img-news**.  
*Увага.* Ім'я параметра не може містити пробіл;
- **Заголовок.** Уводимо заголовок, яким буде підписано додаткове поле при редагуванні ресурсу. Наприклад: **Зображення для сторінки**;
- **Опис.** Супровідний текст, який дасть користувачеві уявлення про те, де буде використовуватися додаткове поле. Опис буде відображатися на сторінці редагування ресурсу під заголовком TV-параметра. Наприклад: **Зображення для ресурсу новини повинне бути розмірами 60 px на 60 px**;
- **Тип параметра.** За допомогою цього поля можна вибрати, якого типу буде додаткове поле для ресурсу. Наприклад: **image**;
- **Візуальний компонент.** За допомогою цього поля можна вибрати HTML-компонент, який буде використовуватися для виведення параметра в шаблоні. Наприклад: **image**.

У блоці **Доступ шаблонів** обираємо шаблони, для яких необхідно створити даний TV-параметр. Після заповнення вказаних полів зберігаємо TV-параметр натисканням кнопки **Зберегти**.

Після виконання вказаних вище дій, перейшовши на сторінку редагування новини побачимо, що на сторінці редагування ресурсу з'явилося ще одне поле, яке розташоване нижче візуального редактора.



Параметри (TV)	
img	<input type="text"/>
<small>[img]</small>	<input type="button" value="Вставити"/>
<small>Зображення для новини</small>	

**Рис. 21.** Вигляд поля TV-параметра у формі редагування ресурсу

### Додавання зображення до ресурсів

При натисканні кнопки **Вставити**, яка розміщена поруч із додатковим полем, відкривається файловий менеджер, що дозволяє вибрати необхідне зображення з папки **assets/images/** або, якщо його там ще немає, то завантажити зображення в цю папку, щоб згодом його обрати.

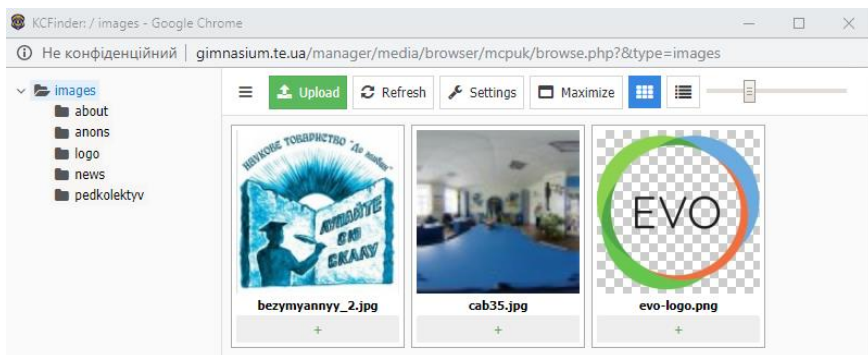


Рис. 22. Вигляд файлового менеджера адміністративної панелі

За допомогою цього файлового менеджера зручно створити дерево каталогів, у які можна розміщати зображення, що в подальшому можуть вставлятися в різних розділах сайту.

**Завдання.** Створіть каталог *news* і в нього помістіть кілька зображень розміром 60 px на 60 px, які оберіть в якості значення TV-параметра для кожного з дочірніх ресурсів сторінки *Новини*.

### Виведення стрічки новин із зображеннями

Після того як усі зображення прикріплені до новин, необхідно вивести блоки з коротким описом новин і зображеннями у стрічці новин.

Для цього у виклику сніпшета **DocLister** необхідно додати параметри **&tvList** і **&tvPrefix**. У параметрі **&tvList** необхідно вказати список назв TV-параметрів, які повинні бути присутні у вибірці, до того ж, якщо таких параметрів кілька, то їх назви необхідно вказати через знак коми. Наприклад: **&tvList='img-news'**. Параметр **&tvPrefix** задає префікс для плейшолдерів, створюваних з імен TV-параметрів. Значення за замовчуванням **tv**. У шаблоні виведення окремого ресурсу в потрібному місці необхідно вказати плейшолдер вигляду **[+tv.img-news+]**

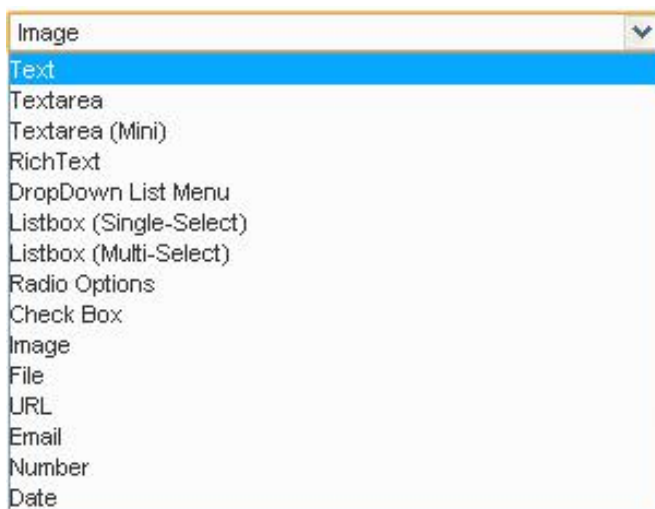
Тоді чанк «*news-article*» виведення окремої новини матиме вигляд:

```
<div id="news">
<a href="[~{+id+}]">img src="[+tv.img-news+]" alt="[+pagetitle+]"</a>
<h2><a href="[~{+id+}]" title="[+pagetitle+]">[+pagetitle+]</a></h2>
<div class="introtext"><span class="date">[+date+]</span> [+introtext+] </div>
<div class="readmore"><a href="[~{+id+}]" title="Перейти до повного тексту новини">Читати
далі </a></div>
</div>
```

Слід звернути увагу на те, що для виведення зображення у стрічці новин використовують назву створеного нами TV-параметра з префіксом **tv**.

*Зауваження.* Коли використовуємо TV-параметр у шаблоні виведення сніппета **DocLister**, то виклик здійснюється за допомогою конструкції вигляду `[+tv.name+]`; якщо ж необхідно вивести вміст додаткового поля безпосередньо на сторінці ресурсу, то конструкція виведення TV-параметра буде мати вигляд `[*name*]`. Це має стосунок і до всіх стандартних полів, таких як `[*pagetitle*]`, `[*longtitle*]`, `[*introtext*]` й інших плейсхолдерів.

Розглянувши можливі значення типів TV-параметра, можна уявити, яку функціональність несе в собі це розширення ModX. Крім додавання зображень, можна прикріплювати до ресурсів додаткові поля таких типів: текстові поля, поля для введення дати, поля для прикріплення файлу, поле для введення URL, для e-mail, чекбокси, радіокнопки, списки, що випадають, текстові поля `textarea` і навіть додаткові поля з візуальним редактором.



**Рис. 23.** Типи TV-параметрів

## **ФОРМА ЗВОРОТНОГО ЗВ'ЯЗКУ В MODX. СНІППЕТ FORMLISTER.**

Одна з основних функцій будь-якого сайту — організація взаємозв'язку між власником сайту й відвідувачами. Важко уявити, що цей взаємозв'язок буде повним, якщо не буде надано відвідувачеві можливість виразити свою точку зору або просто зв'язатися з того чи іншого приводу із власником. Зворотній зв'язок — дуже важливий елемент для сайту будь-якого типу. Саме завдяки зворотному зв'язку спілкування може бути повноцінним і відбуватися у двох напрямках: «власник — відвідувач» і «відвідувач — власник». Ураховуючи все це, розглянемо питання створення на сайті форми зворотного зв'язку.

Спочатку необхідно описати всі вимоги до форми, що буде створена. Перш за все, це наявність наступних полів з такими вимогами:

- **ім'я** — поле, обов'язкове для заповнення;
- **електронна пошта** — обов'язкове для заповнення поле, яке повинне перевірятися ще й на правильність уведення e-mail;
- **повідомлення** — поле, обов'язкове для заповнення.

Також необхідно реалізувати оповіщення відвідувачів про помилки введення інформації при заповненні форми. Ця вимога до форми зворотного зв'язку — наявність захисту від спаму у вигляді поля для введення тексту зі згенерованого зображення (капчі). Дана форма повинна надсилатися за допомогою Ajax-методу, тобто без перезавантаження сторінки.

Для створення форми зворотного зв'язку в ModX існує кілька сніпсетів, однак розглянемо сніпсет **FormLister**, що входить до переліку сніпсетів, які встановлюються разом із дистрибутивом CMS ModX Evo.

Виклик сніпсета **FormLister** нічим не відрізняється від виклику інших сніпсетів. Основні параметри, які обов'язково потрібно знати при роботі в ModX із цим сніпсетом, розглянемо на такому прикладі:

```
[!FormLister?
&formid=`form-callme`
&rules={`
  "name":{
    "required":"Не введено ім'я"
  },
  "email":{
    "required":"Не введено Email",
    "email":"Не вірно введено Email"
  },
  "message":{
    "required":"Не введено текст повідомлення"
  }
}`
&formTpl=`form-callme-tpl`
&messagesOuterTpl=`@CODE:<div class="alert alert-danger"
role="alert">[+messages+]</div>`
&errorTpl=`@CODE:<span class="help-block">[+message+]</span>`
&errorClass=`has-error`
&requiredClass=`has-warning`
&to=`manager@fizmat.tnpu.edu.ua`
&subject=`Повідомлення з сайту`
&reportTpl=`form-callme-report`
!]
```

У цій конструкції виклику сніпета використовують такі параметри `FormLister`:

- **&formid** — унікальний ідентифікатор форми. Це значення має обов'язково вказуватися в шаблоні форми в прихованому полі або як значення `id` у тезі `form`;
- **&to** — значенням цього параметра є розділений комами список електронних адрес можливих одержувачів листа;
- **&formTpl** — параметр, який передає сніпету назву чанка, що буде використовуватися в якості шаблону форми;
- **&reportTpl** — параметр, який передає сніпету назву чанка, що буде використовуватися в якості шаблону для відправлення даних на пошту;
- **&messagesOuterTpl** — шаблон повідомлень обробника форми. У шаблоні виводяться групи повідомлень (`messages`, `required`, `error`);
- **&errorTpl** — шаблон для виведення повідомлень валідатора;
- **&rules** — масив з правилами для валідації;
- **&subject** — у цей параметр можна вписати текст, який буде використаний у якості теми електронного листа.

Шаблон форми, який знаходиться в чанку `form-callme-tpl`, може мати такий вигляд:

```
<form class="subset-form row m0" method="post" id="contact-form">
  <input type="hidden" name="formid" value="form-callme">
  <div class="form-group">
    <input type="text" class="form-control" name="name"
placeholder="Ім'я*" value="[+name.value]" data-error="Не введено
ім'я" required>
    <div class="help-block with-errors"></div>
  </div>
  <div class="form-group">
    <input type="email" class="form-control" name="email"
placeholder="Email*" value="[+email.value]" data-error="Не введено
email" required>
    <div class="help-block with-errors"></div>
  </div>
  <div class="form-group">
    <textarea class="form-control message" name="message"
placeholder="Повідомлення*" data-error="Не введено текст повідом-
лення" required>[+message.value]</textarea>
    <div class="help-block with-errors"></div>
  </div>
  <div class="form-response">[+form.messages+</div>
  <div class="result"><button class="submit" type="submit">Наді-
слати</button></div>
</form>
```

Чанк **form-callme-report** може мати такий вміст:

```
<p>Ім'я: [+name.value+]</p>
<p>Email: <a href="mailto:[+email.value+]">[+email.value+]</a></p>
<p>Повідомлення: [+message:strip_tags:nl2br+]</p>
```

У цьому шаблоні вся введена користувачем інформація передається за допомогою плейсхолдерів вигляду [+name.value+]. При цьому назва кожного плейсхолдера збігається зі значенням атрибута **name** відповідного поля в HTML-шаблоні (name="name", name="email", name="message").

Список правил для валідації правильності заповнення елементів форми задається у вигляді масиву. Ключем є ім'я поля, а значенням — масив правил валідації. Правило валідації є методом класу-валідатора. У масиві правил ключем є ім'я правила (назва методу валідації), значенням може бути або рядок з повідомленням про помилку валідації правила, або ж масив з описом. У цьому масиві в ключі **params** задаються необхідні для валідації значення, а в ключі **message** задається рядок з повідомленням про помилку.

```
{
  "ім'я поля 1": {
    "правило 1" : "повідомлення про помилку",
    "правило 2" : "повідомлення про помилку "
  },
  "ім'я поля 2": {
    "правило 1" : "повідомлення про помилку",
    "правило 2" : {
      "params" : значення,
      "message" : "повідомлення про помилку"
    }
  }
}
```

Якщо потрібно реалізувати перевірку тільки заповнених полів, то перед іменем поля в списку правил потрібно додати знак оклику. У цьому випадку, якщо значення поля порожнє, правила будуть проігноровані.

Відправлене за допомогою форми зворотного зв'язку повідомлення має прийти на електронну адресу, яку необхідно вказати при виклику сніпета **FormLister** за допомогою параметра **&to**. На локальному комп'ютері (наприклад, з локальним web-сервером Denwer), відправлені листи можна знайти у каталозі **z:\tmp\sendmail\**. У цей каталог будуть відсилатися листи, відправлені за допомогою поштових скриптів на локальному комп'ютері. Ім'я листа буде сформовано з дати відправлення. Із web-сервером ХАМРР все дещо складніше. Для того щоб бачити відправлені листи, необхідно виконати такі кроки:

1. У каталозі **sendmail (C:\xampp\sendmail)** потрібно створити файл **sendmail.php** з кодом:

```
<?php
```



```

define('DIR', 'c:/xampp/tmp/sendmail/');
$stream = '';
$fp = fopen('php://stdin', 'r');
while($t=fread($fp,2048)){
    if( $t==chr(0) ) break;
    $stream .= $t;
}
fclose($fp);
$fp = fopen(mkname(), 'w');
fwrite($fp,$stream);
fclose($fp);
function mkname($i=0){
    $fn = DIR.date('Y-m-d_H-i-s_').$i.'.eml';
    if ( file_exists($fn) )return mkname(++$i);
    else return $fn;
}
?>

```

2. У файлі **php.ini** (C:\xampp\php) замінити стрічку:

```
sendmail_path = "C:\xampp\sendmail\sendmail.exe -t"
```

на стрічку:

```
sendmail_path = C:\xampp\php\php.exe c:\xampp\sendmail\sendmail.php
```

Стрічка повинна бути розкоментована (витерти символ крапки з комою ";" на початку стрічки).

3. Створити каталог **sendmail** у каталозі tmp (C:\xampp\tmp\sendmail).

4. Перезапустити сервер Apache.

Після виконання цих пунктів усі надіслані листи будуть потрапляти в каталог C:\xampp\tmp\sendmail.

## РЕАЛІЗАЦІЯ ГАЛЕРЕЇ ЗОБРАЖЕНЬ. МОДУЛЬ SIMPLEGALLERY.

Для реалізації галереї в ModX розроблено багато готових рішень у вигляді додаткових модулів і плагінів: maxigallery, Easy 2 Gallery, MiniGallery, EvoGallery тощо. Але всі вони призначені для вузького застосування, тому їх використання накладає певні обмеження на розробників. Найбільш вдалим на сьогодні рішенням є модуль SimpleGallery, який є простим у встановленні та легким у налаштуванні, до того ж вимагає від розробника мінімум часу для адаптації під свої потреби.

SimpleGallery є динамічною галереєю, розробленою як модуль під CMS ModX, що дозволяє швидко і легко створювати галереї зображень, пов'язуючи їх з будь-якими сторінками на сайті. Галереї можуть виводитися у різний спосіб із використанням шаблонів системи. SimpleGallery не створює автономні галереї, які не пов'язані з конкретною сторінкою. Кожне завантажене фото повинне бути пов'язане зі сторінкою у дереві ресурсів сайту.

### Можливості SimpleGallery:

- управління зображеннями на сторінці редагування ресурсу в окремій вкладці;
- пагінація з можливістю вибору кількості картинок на сторінці;
- завантаження безлічі картинок — можна кнопкою, можна перетягуванням (використовується jquery.fileapi)
- зменшення картинок при завантаженні до значень, вказаних у конфігурації плагіна;
- транслітерація і присвоювання унікальних імен при завантаженні файлів (як в KCFinder);
- картинку можна видаляти по одній, можна виділити (ctrl + click, shift + click, ctrl + a) і видалити кілька;
- картинку можна переміщати в початок / кінець поточного галереї, переміщати в іншу галерею;
- сортування перетягуванням (використовується бібліотека Sortable);
- редагування у вікні подвійним кліком (як в ms2Gallery);
- інтерфейс можна перевести на різні мови, в наявності російська та англійська;
- кешування картинок в адміністративній панелі браузером для прискорення роботи;
- система подій для обробки елементів галереї, у тому числі реалізована можливість примусової обробки («оновити прев'ю»);
- можливість змінювати інтерфейс і логіку роботи плагіна без правок початкових кодів.

### ***Встановлення SimpleGallery***

Для розширення функціональних можливостей CMS ModX Evolution необхідно вміти встановлювати додаткові сніпшети, плагіни та модулі, які дозволяють розробнику без зайвих проблем за рахунок використання готових рішень розв'язувати типові та нетипові завдання, які можуть виникати в процесі розробки. Однак у розглядуваній версії ModX Evo є вбудований модуль **Extras** (встановлюється разом зі встановленням CMS), який дозволяє спростити процес встановлення нових додаткових пакетів розширень із репозитарію ModX.

Для встановлення **SimpleGallery** необхідно перейти у модуль **Extras** за посиланням **Модулі – Extras**, де у стрічці пошуку ввести назву необхідного модуля і в результатуючому переліку на іконці **SimpleGallery** натиснути кнопку **Install**.

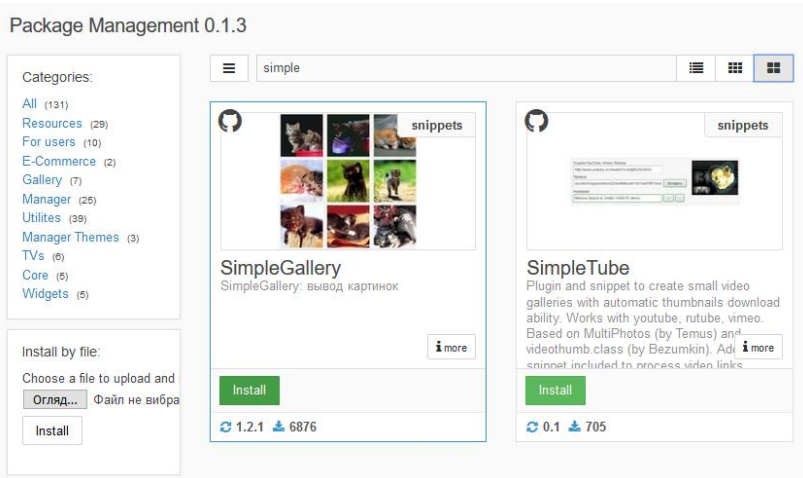


Рис. 24. Вигляд вікна модуля Extras

Після вдалого встановлення повинні з'явитися такі плагіни: **sgThumb**, **SimpleGallery**, сніппети: **sgController**, **sgLister**, **sgThumb**. Зауважимо, що для повноцінного функціонування модуля необхідно, щоб був встановлений сніппсет **DocLister** (у нашій збірці він встановлений разом зі встановленням CMS).

### Налаштування SimpleGallery

Для здійснення налаштування плагіна SimpleGallery, що відповідає за параметри галереї в адміністративній панелі, необхідно у вкладці **Конфігурація** заповнити такі поля:

- **Tab name** — назва вкладки;
- **Controller class** — ім'я доступного класу-контролера;
- **Templates** — ID шаблонів ресурсів, які повинні містити галерею (вказуються через кому, наприклад: «10,11,12»);
- **Documents** — те ж саме, але для окремих ресурсів (вказуються через кому, наприклад: «5,8,9»);
- **Ignore Documents** — ID ресурсів, що не повинні містити галерею (вказуються через кому, наприклад: «1,15,16»);
- **Roles** — ID дозволених ролей;
- **Galleries folder** — каталог для зберігання зображень галерей;
- **Thumbs cache folder** — каталог для зображень прев'ю перегляду (тих, що в адміністративній панелі);
- **Thumbs width** — ширина зображень прев'ю перегляду;
- **Thumbs height** — висота зображень прев'ю перегляду;

- **Custom thumb options** — тут за необхідності можна вказати параметри `phpthumb` для генерації зображень прев'ю перегляду в адміністративній панелі; плейсхолдери `[+ w +]` і `[+ h +]` містять значення параметрів `Thumbs width` і `Thumbs height`;
- **Client Resize** — попередня обробка зображень на стороні клієнта.

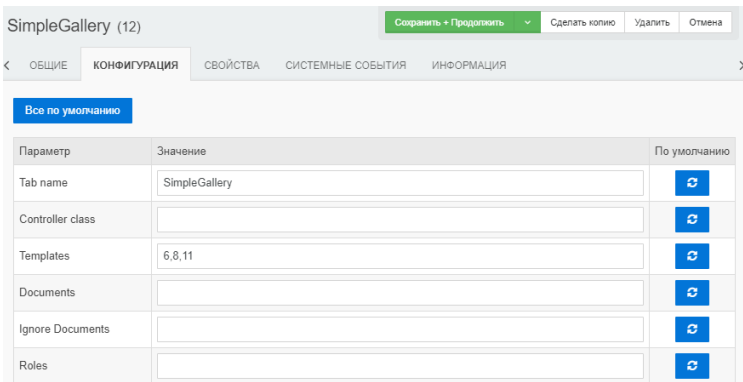


Рис. 25. Вигляд вікна налаштування конфігурації плагіна SimpleGallery

### Використання SimpleGallery

Для завантаження зображень у галерею в адміністративній панелі переходимо до ресурсу, що повинен містити галерею зображень. Якщо все вірно налаштовано (ID ресурсу входить до переліку в полі **Documents** або ID шаблону ресурсу входить до переліку у полі **Templates**), то у вікні редагування ресурсу буде міститися вкладка **SimpleGallery** (або та назва, що вказана в полі **Tab name** у конфігурації плагіна).

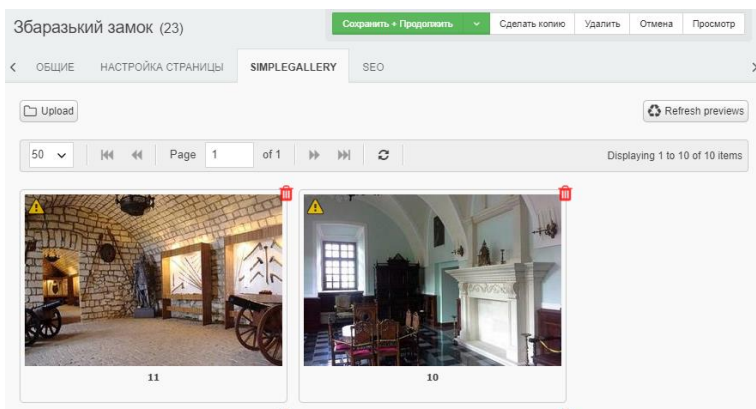
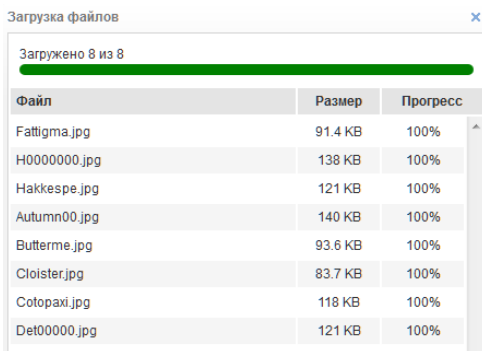


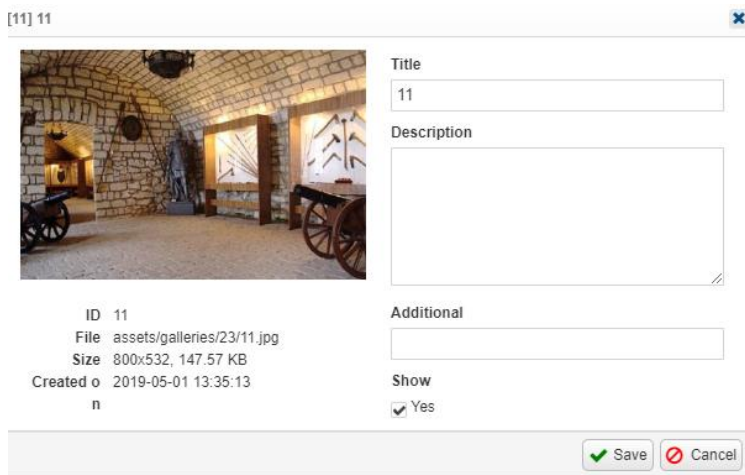
Рис. 26. Вигляд вкладки SimpleGallery вікна редагування ресурсу

У вкладці **SimpleGallery** клікаємо по кнопці **Upload**, і у вікні вибору зображень, що з'явиться, обираємо потрібні файли зображень, після чого натискаємо кнопку **Відкрити**, після чого з'явиться вікно, що відображатиме процес завантаження файлів.



**Рис. 27. Вигляд вікна завантаження файлів зображень галереї SimpleGallery**

Окрім завантаження самих картинок, існує можливість задавати їх назви і опис. Для цього необхідно здійснити подвійний клік лівою клавшею мишки по потрібному зображенні й у вікні, що з'явиться, заповнити відповідні поля.



**Рис. 28. Вигляд вікна налаштування параметрів завантажених зображень**

Для виведення на сторінці ресурсу галереї зображень у відповідному шаблоні в потрібному місці необхідно викликати сніппет **sgLister**:

`[[sgLister?`

```
&thumbSnippet=`phthumb`
&thumbOptions=`w=150&h=150&z=1`
&tpl=@CODE:<a href="[+sg_image+]"></a>`]],
```

де **&thumbSnippet** — ім'я сніпета, який поверне посилання на прев'ю, наприклад, **phthumb**; **&thumbOptions** — параметри для створення прев'ю в тому вигляді, у якому їх прийме вказаний сніпет; **&tpl** — HTML-код шаблону для виведення окремого зображення галереї (за бажання можна винести в окремий чанк).

Для виведення списку галерей потрібно використати сніпет **sgController**. Його виклик має такий вигляд:

```
[[sgController?
&count=`1`
&depth=`1`
&ownerTPL=@CODE:[+dl.wrap+]`
&tpl=@CODE:<h2>[+pagetitle+] <span
class="badge">[+count+]</span></h2>[+images+]`
&sgDisplay=`4`
&sgOuterTpl=@CODE:<div class="row">[+wrapper+]</div>`
&sgRowTpl=@CODE:
<div class="col-lg-3 col-md-4 col-xs-6 thumb">
  <a class="thumbnail" href="[+url+]">
    
  </a>
</div>
`
&thumbSnippet=`sgThumb`
&thumbOptions=`400x300`
&orderBy=`menuindex ASC`
&sgOrderBy=`sg_index DESC`
]]
```

З описом списку використовуваних параметрів можна ознайомитися за посиланням

[http://docs.evo.im/04\\_extras/simplegallery/vyvod\\_spiska\\_galerey.html](http://docs.evo.im/04_extras/simplegallery/vyvod_spiska_galerey.html)

Модуль **SimpleGallery** вельми гнучкий інструмент, крім складних галерей з його допомогою можна додавати різні слайдери на сайт, а також прив'язувати красиві скрипти.

## РОБОТА З МОДУЛЕМ DOC MANAGER

За допомогою модуля **Doc Manager** існує можливість швидко редагувати властивості всіх створених ресурсів, змінювати дату публікації й шаблон

для кожної сторінки. За допомогою цього модуля розробник має можливість заощадити багато часу, коли в дереві буде створено дуже багато ресурсів.

Наприклад, щоб за допомогою модуля скасувати показ у меню сторінок переходимо у вкладку: **Модулі** → **Doc Manager** → **Інші властивості**. Зі списку, що випадає, **Доступні властивості** обираємо пункт **Показувати/Не показувати в меню**, після чого нижче встановлюємо чекбокс **Не показувати в меню**. Далі слід вказати діапазон ID ресурсів, які необхідно скасувати для показу у меню. Після вказівки ID ресурсів слід натиснути кнопку **Відправити**.

## **ПЕРЕНЕСЕННЯ ГОТОВОГО MODX САЙТУ НА ХОСТИНГ**

Перенесення сайту на хостинг зводиться до кількох простих кроків:

- вибір і замовлення хостингу й доменного імені;
- копіювання файлів сайту в певний каталог на сервері;
- створення бази й користувача бази даних у хостера з наступним копіюванням у неї вмісту бази з локального комп'ютера;
- редагування конфігураційного файлу для підключення створеної бази даних до сайту;
- скидання налаштувань і кеша, які залишилися в системі керування з локального комп'ютера.

## ДОДАТКИ

### Додаток 1. Спеціальні теги ModX

[[**site\_name**]] — виводить заголовок сайту. Зазвичай використовують у заголовках HTML-сторінок у тезі <title>.

[[**base\_url**]] або [[**site\_url**]] — два теги ідентичні між собою. Ці конструкції дозволяють виводити URL сайту.

[[**\*pagetitle\***]] — виводить вмістиме поля **Заголовок**, яке заповнюється на сторінці створення/редагування ресурсу.

[[**\*longtitle\***]] — виводить вмістиме поля **Розширений заголовок**.

[[**\*description\***]] — виводить вмістиме поля **Опис**. Це поле як правило використовують для виведення вмістимого в META-тезі description.

[[**\*introtext\***]] — виводить вмістиме поля **Анотація**. Це поле зазвичай використовують при створенні новин, заміток в блозі і т. п. для організації сторінок з коротким описом заміток.

[[**\*content\***]] — основний вміст сторінок. Конструкція виводить довільний текст або HTML-код, написаний або відредагований у візуальному редакторі.

[[**\*id\***]] — виводить ідентифікатор ресурсу.

[[**\*alias\***]] — виводить псевдонім ресурсу.

[[~**ідентифікатор**~]] — виводить URL адресу ресурсу, ідентифікатор якого вказано. Наприклад, якщо ID сторінки «Новини» — 4, а псевдонім цієї сторінки — **news**, то конструкція [[~4~]] виведе URL цієї сторінки з новинами.

[[**\*pub\_date\***]] — дата публікації ресурсу.

[[**\*unpub\_date\***]] — дата завершення публікації.

[[**\*createdby\***]] — ідентифікатор користувача, що створив ресурс.

[[**\*createdon\***]] — дата створення ресурсу.

[[**\*editedby\***]] — ідентифікатор користувача, що редагував ресурс.

[[**\*editedon\***]] — дата редагування ресурсу.

[[**\*contentType\***]] — тип вмістимого (наприклад, text/html).

[[**\*type\***]] — тип (ресурс, папка або посилання).

[[**\*published\***]] — чи опубліковано ресурс (1|0).

[[**\*parent\***]] — номер (ID) батьківського ресурсу.

[[**\*isfolder\***]] — чи є ресурс папкою (1|0).

[[**\*richtext\***]] — чи використовується при редагуванні сторінки візуальний редактор.

[[**\*template\***]] — номер (ID) шаблону ресурсу.

[[**\*menuindex\***]] — порядковий номер відображення в меню.

[[**\*searchable\***]] — чи доступний ресурс для пошуку (1|0).

[[**\*cacheable\***]] — чи кешується ресурс (1|0).

[[**\*deleted\***]] — ресурс знищено (1|0).

[[**\*deletedby\***]] — ідентифікатор користувача, що знищив ресурс.

[[**\*menutitle\***]] — заголовок меню, якщо такий існує.



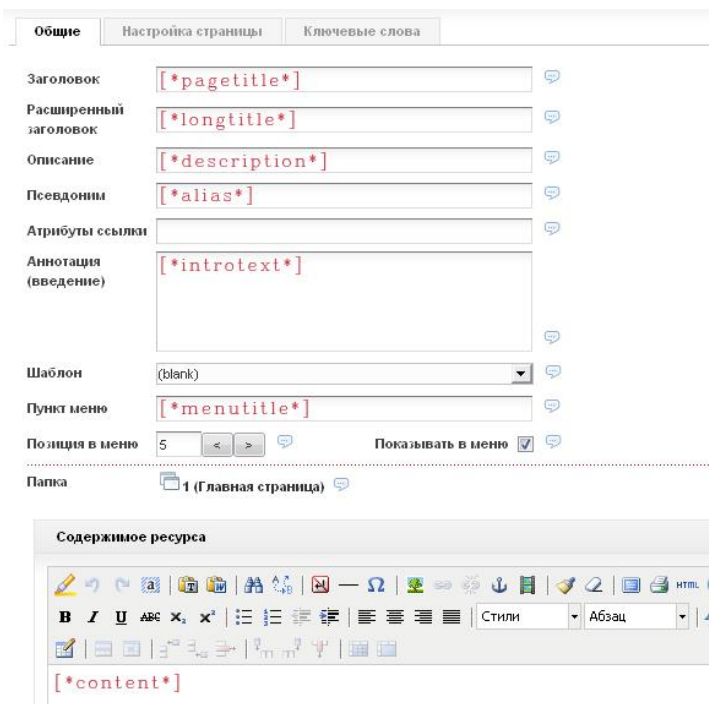


Рис. 29. Плейсхолдеры вікна редагування ресурсів

**[\*donthit\*]** — відслідковує кількість відвідувань (1|0).

**[\*haskeywords\*]** — ресурс містить ключові слова (1|0).

**[\*hasmetatags\*]** — ресурс має META-теги (1|0).

**[\*hidemenu\*]** — документ не відображається в меню (1|0).

**[(modx\_charset)]** — виводить назву кодування, що використовується.

**[^qt^]** — виводить час запиту до бази даних.

**[^q^]** — виводить кількість запитів до бази даних.

**[^p^]** — виводить час роботи PHP-скриптів.

**[^t^]** — виводить загальний час генерації сторінки.

**[^s^]** — виводить джерело вмістимого (база чи кеш).

### Додаток 2. Параметри та плейсхолдери сніпнета DLMenu

**[+wrap+]** — виведення дочірніх документів (в api-режимі — children); якщо значення плейсхолдера — масив, то він перетворюється в рядок із застосуванням відповідних шаблонів, якщо рядок — залишається без змін.

**[+ classNames +]** — список класів, доступних в шаблоні (тільки імена).

**[+ classes +]** — список класів, доступних в шаблоні (включаючи class = "").

**[+ level +]** — рівень документа.

[+ **maxLevel** +] — якщо встановлено, то вказує на те, що документ міститься у самому низу гілки, до якої належить.

[+ **iteration** +] — порядковий номер документа в групі.

[+ **\_display** +] — загальна кількість документів в групі.

[+ **here** +] — якщо встановлено, то документ є поточним документом.

[+ **active** +] — якщо встановлено, то документ є активним документом.

[+ **state** +] — якщо заданий параметр **hideSubMenus**, то плейсхолдер містить значення **open** для розгорнутої гілки і **closed** — для згорнутої.

[+ **title** +] — назва документа, аналогічне до поля **menutitle**, якщо воно не порожнє, або **pagetitle**.

[+ **url** +] — посилання на документ.

[+ **count** +] — кількість безпосередніх нащадків.

[+ **\_renderRowTpl** +] — якщо встановлено, то його значення буде використано як шаблон при виведенні документів.

[+ **\_renderOuterTpl** +] — якщо встановлено, то його значення буде використано як шаблон при виведенні обгортки дочірніх документів.

**&documents** — список ID документів, які утворюють однорівневе меню. Щоб вивести в тій послідовності, у якій вони перераховані, слід використовувати параметр **sortBy** зі значенням **doclist**.

**&parents** — список ID документів, від яких будується багаторівневе меню, через кому. Якщо у зазначених документів різні батьки, то буде побудовано кілька меню. Щоб вивести в тій послідовності, у якій перераховані батьки, слід використовувати параметр **sortBy** зі значенням **doclist**.

**&maxDepth** — максимальна кількість рівнів в меню;

**&showParent** — якщо значення параметра дорівнює 1, то будуть показані документи, зазначені в параметрі **parents**.

**&hideSubMenus** — якщо значення параметра дорівнює 1, то неактивні гілки меню не показуються.

**&outerTpl** — обгортка всього меню

`(@CODE:<ul [+classes+]>[+wrap+]</ul>)`.

**&rowTpl, &rowHereTpl** — шаблон для виведення пункту меню першого рівня без нащадків. Для поточного документа може бути заданий шаблон **rowHereTpl**

`(@CODE:<li [+classes+]><a href="[+url+]">[+title+]</a></li>)`.

**&parentRowTpl, &parentRowHereTpl, &parentRowActiveTpl** — шаблон для виведення документа, у якого є нащадки. Для поточного документа може бути заданий шаблон **parentRowHereTpl**, для активного — **parentRowActiveTpl**

`(@CODE:<li [+classes+]><a href="[+url+]">[+title+]</a>[+wrap+]</li>)`.

**&innerTpl** — обгортка блоку дочірніх документів. За замовчуванням — значення параметра **outerTpl**.

**&innerRowTpl, &innerRowHereTpl** — шаблон для виведення дочірнього документа. Для поточного документа може бути заданий шаблон **innerRowHereTpl**.

### *Додаток 3. Параметри сніпнета DocLister*

**&idType** — тип вибірки документів. Можливі значення — **parents, documents**. Значення за замовчуванням — **parents**.

**&parents** — вибірка документів на підставі списку батьківських документів. Можливі значення — значення ID батьківських документів, розділені комою. Значення за замовчуванням — ID сторінки, на якій викликається сніпсет.

**&documents** — вибірка довільних документів. Якщо використовується параметр **parents\_**, то документи, перераховані в цьому параметрі, будуть просто підмішані в результат і подібні до наступних правил вибірки (фільтрація, сортування). Можливі значення — значення ID документів, розділені комою.

**&ignoreEmpty** — дозволяє зробити вибірку всіх записів з таблиці, якщо параметр **documents** не заданий. Параметр **idType** в цьому випадку повинен бути **documents**. Можливі значення — 1 або 0. За замовчуванням — 0.

**&display** — максимальна кількість документів у вибірці.

**&depth** — глибина вибірки з використанням параметра **parents**. Можливі значення — ціле число, яке більше або дорівнює нулю. Значення за замовчуванням — 0.

**&offset** — кількість документів, що пропускаються з початку списку. Перевизначається при використанні пагінації. Якщо ж потрібно завжди пропускати N документів, то необхідно використовувати параметр **start**. Можливі значення — ціле число, яке більше або дорівнює нулю. Значення за замовчуванням — 0.

**&start** — число документів, що пропускаються з початку вибірки. Додається до значення **offset**, що автоматично встановлюється при пагінації.

**&total** — максимальна кількість документів, що відображаються на одній сторінці у вибірці. Можливі значення — ціле число, яке більше або дорівнює нулю. Значення за замовчуванням — 0.

**&addWhereList** — додаткові умови вибірки документів. Будь-який рядок, що задовольняє умови рядка для підстановки в WHERE блок SQL-запиту.

**&showParent** — вилучення документів, з яких робилася вибірка з використанням параметра **parents**.

**&selectFields** — імена полів, що включаються до вибірки.

**&groupBy** — групування результатів за деяким полем.

**&sortType** — значення **none** визначає автоматичне сортування за правилами MySQL (як правило, за **primary key**).

**&orderBy** — єдина стрічка сортування (як мінімум сукупність параметрів **sortBy** і **sortDir**, але має більший пріоритет). Для сортування у випадковому порядку значення параметра **orderBy** буде **RAND()**.

**&sortBy** — впорядковано за без напрямку сортування. Можливі значення — будь-який рядок, що задовольняє правилам побудови параметра ORDER BY в SQL запиті. Імена TV-параметрів вказуються як `€`.

**&order** — напрямок сортування. Можливі значення — ASC, DESC. Значення даного параметра може бути перевизначити значенням параметра **sortDir**. Значення за замовчуванням — DESC.

**&sortDir** — синонім параметра **order**, але має більший пріоритет. Якщо встановлені два параметри: **order** і **sortDir**, то буде використовуватися значення параметра **sortDir**. Можливі значення — ASC, DESC. Значення за замовчуванням — DESC (або визначається в контролері).

**&urlScheme** — схема генерації URL. Можливі значення — схеми, доступні в MODX Evolution (відносні, http, https, full). Значення за замовчуванням — порожньо (відносні).

**&dateSource** — поле документа, в якому розташовується дата. Можливі значення — назва поля в таблиці. Якщо в якості значення було вказано рядок, відмінний від `createdon` і значення даного поля в базі дорівнює 0, то тоді все одно береться значення з `createdon`. Для прикладу: якщо використовується відкладена публікація для деяких документів і вказується сортування по полю `pub_date`.

**&dateFormat** — правила форматування дати **format** для PHP функції `strftime`. Як джерело дати використовується параметр **dateSource**. Крім цього враховується зсув дати на сервері (див. системний параметр `server_offset_time`). Таким чином, можна використовувати персоналізовану підстановку часу залежно від часового поясу користувача. Значення за замовчуванням — % d.% B.% Y% H:% M

**&summary** — правила обробки текстів для формування короткого опису. Завантажує екстендер `summary`. У контролері `site_content` є додаткове правило для оброблюваного тексту: за замовчуванням на обробку відправляється поле **content**. Але якщо поле **introtext** не порожнє, то текст саме з цього поля буде переданий на додаток **summary**. Аналогічним чином собі веде і контролер **onetable**. Можливі значення — рядок, сформований за правилами екстендера `summary`: дія1: параметр1, дія 2: параметр2: параметр2Б, дія 3.

**&introField** — ім'я поля для джерела короткого опису тексту з **contentField**. Використовується тільки якщо завантажений екстендер `summary`.

**&contentField** — ім'я поля, в якому зберігається основний текст документа. Використовується тільки якщо завантажений екстендер `summary`.

**&ownerTPL** — шаблон, в який обертається список результатів. Якщо потрібних результатів не виявлено, то за замовчуванням результат з шаблоном **noneTPL** теж повертається в цей шаблон. Значення за замовчуванням — порожньо.

**&tpl** — шаблон. Приклад:

```
&tpl=`@CODE:
```

```
<li>
  
  <a href="[+url+]">[+title+] ([+id+])</a>
</li>
```

**&paginate** — виведення даних з пагінацією. Можливі значення: **offset** — пагінація в стилі Ditto; **pages** — пагінація з прострілами. Значення за замовчуванням — **pages**.

**&tplFirstP** — шаблон для вставки посилання "перша сторінка".

**&tplLastP** — шаблон для вставки посилання "остання сторінка".

**&tplNextP** — шаблон для вставки посилання "наступна сторінка". Значення за замовчуванням: @CODE: <a href="[+link+]">'. \$ This-> DocListe-> getMsg ( 'paginate.next', 'Next'). '> </a>

**&tplPrevP** — шаблон для вставки посилання "попередня сторінка". Значення за замовчуванням: @CODE: <a href="[+link+]"> <'. \$ This-> DocListe-> getMsg ( 'paginate.prev', 'Prev'). '</a>

**&tplPage** — шаблон для вставки номера сторінки в пагінатор. Значення за замовчуванням: @CODE: <a href="[+link+]" class="page"> [+ num +] </a>

**&tplCurrentPage** — шаблон оформлення поточної сторінки в пагінаторі. Значення за замовчуванням: @CODE: <b class = "current"> [+ num +] </ b>

**&tplWrapPaginate** — шаблон контейнер для обгортки сторінок пагінації. Значення за замовчуванням: @CODE: <div class = "[+ class +]"> [+ wrap +] </ div>

**&pageLimit** — кількість сторінок відображається в пагінаторі. Можливі значення — ціле число більше нуля. Значення за замовчуванням — 1.

**&pageAdjacents** — максимальна кількість сторінок зліва і справа щодо поточної сторінки. Можливі значення — ціле число більше нуля. Значення за замовчуванням — 4.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Antano Solar John. MODx Web Development. Second Edition. ПАСКТ.
2. Шпак Ю. Web-разработка средствами MODx, 2012 год, Пресс, 400 с.
3. Разработка современного портала вуза на базе CMS MODx. Лукиных Н. Н., Сироткин А. Ю. Психолого-педагогический журнал Гаудеамус. 2012. Т. 2. № 20. С. 204.
4. Горнаков С. Г. Осваиваем популярные системы управления сайтом (CMS). М: ДМК Пресс, 2009. 336 с.
5. MODX CMS, система управление содержимым, CMS системы, управление сайтом. Режим доступа: <https://modx.ru/o-sisteme-modx/>
6. Школа MODX Evolution. Режим доступа: <http://i-gu.ru/>
7. MODX EVO by Dmi3yy. Режим доступа: <http://modx.com.ua/>
8. Документация по Evolution CMS. Режим доступа: <http://docs.evo.im/>

## ЗМІСТ

Встановлення та налаштування локального сервера XAMPP .....	3
Встановлення XAMPP .....	3
Створення віртуального хоста.....	5
Створення бази даних MySQL .....	5
CMS MODX Evolution .....	6
Встановлення MODX Evolution .....	6
Налаштування ModX Evolution.....	8
Вкладка «Сайт» .....	8
Вкладка «Дружні URL» .....	10
Вкладки «Користувачі» і «Безпека» .....	11
Вкладка «Інтерфейс і представлення».....	11
Вкладка «Файл-Менеджер».....	12
Інтеграція шаблону в систему управління.....	12
Налаштування шаблону, розбиття на чанки .....	15
Дерево ресурсів і створення сторінок .....	17
Вкладка «Загальні».....	18
Вкладка «Налаштування сторінки» .....	19
Реалізація динамічного меню .....	20
Створення стрічки новин. Сніппет DocLister .....	22
TV-параметри ModX. Додавання зображень до ресурсів. ....	26
Створення TV-параметра.....	27
Додавання зображення до ресурсів .....	27
Виведення стрічки новин із зображеннями .....	28
Форма зворотного зв'язку в ModX. Сніппет FormLister. ....	29
Реалізація галереї зображень. Модуль SimpleGallery. ....	33
Встановлення SimpleGallery .....	34
Налаштування SimpleGallery.....	35
Використання SimpleGallery .....	36
Робота з модулем Doc Manager.....	38
Перенесення готового ModX сайту на хостинг .....	39
Додатки .....	40
Додаток 1. Спеціальні теги ModX .....	40
Додаток 2. Параметри та плейсхолдери сніппета DLMenu .....	41
Додаток 3. Параметри сніппета DocLister .....	43
Список використаних джерел.....	46

**Навчальне видання**

*Вельгач Андрій Володимирович*

# **СТВОРЕННЯ САЙТУ ЗА ДОПОМОГОЮ CMS MODX EVO**

Формат 60×84/16. 2,79 ум. др. арк., 2,46 обл.-вид. арк.  
Тираж 300. Замовлення № 19-059.