

## Порівняльний аналіз найпоширеніших фреймворків для PHP у сфері веб-розробок

У статті стисло аналізуються найпоширеніші фреймворки в галузі інформаційних технологій за параметрами швидкодії, розміру дистрибутиву, зручність у використанні та ін. Проводиться порівняння вихідного коду на прикладах вибірки з БД, валідації даних, контролю кешу.

**програмування, Веб скрипти back-end, фреймворк, каркас, структура MVC, PHP, CakePHP, CodeIgniter, Symfony**

**Вступ.** Ми говоримо, що живимо в освіченому суспільстві. Освічене суспільство – значить суспільство що має вільний доступ до інформації. Основним засобом зв'язку так чи інакше є комунікації за допомогою Інтернет. А основним джерелом інформації є Інтернет сайти.

Інтернет сайт – технічно являє собою машину, що підключена до мережі, і на яку встановлено спеціальне програмне забезпечення – веб сервер.

Веб сервер обробляє запити що надходять до машини, через мережу (запити надсилає веб браузер кінцевого користувача). Визначає який саме скрипт треба викликати і з якими параметрами. Скрипт на виході видає якийсь текст, в залежності від заголовку цього тексту може бути зображенням, сторінкою, текстом, виконуємим файлом, взагалі будь чим. Коли цей текст доходить до браузера він його певним чином обробляє і показує користувачу.

Веб скрипти back-end'у (ті що виконуються на стороні серверу) можна писати на спеціальних мовах, таких як: Java, Basic .Net, C# . Net, Ruby on Rails, Python, Perl, PHP, та деяких інших. Найпоширенішим серед них є PHP [1-3]. Для кожної з цих мов є спеціальні «заготовки», як правило написані товариствами і вільно розповсюджені. Такі «заготовки» називаються фреймворками.

Задачею цієї статті є розгляд і порівняльний аналіз 3х найпоширеніших фреймворків для PHP (CakePHP[4], CodeIgniter[5] та Symfony[6]), за такими параметрами, як: швидкодія, розмір дистрибутиву та зручність у використанні.

Фреймворки задають проекту певну структуру. Містять написані класи та функції для якихось рутинних задач. Найпоширенішим типом структури є MVC (Model View Controller) – система обробки запиту розбивається на 3 частини: контролер, модель, відображення. Маршрутизатор обробляє запит що надходить до серверу і визначає якому саме контролеру передати керування і з якими параметрами. Функція у контролері виконує якийсь код і може викликати модель для роботи з даними. Модель взагалі призначена для роботи з даними. Найчастіше це значить що саме в моделі буде знаходитись код для роботи з БД, для збереження завантажених файлів, а також математичних розрахунків. Після того як контролер відпрацює він передає результати своєї роботи на відображення.

Відображення. Тут знаходиться верстка сторінки в яку вставляються дані що прийшли від контролеру. Відображення може також використовувати «хелпери». В різних фреймворках мають різне смислове забарвлення, втім, в цілому вони будують якісь часто повторювані елементи дизайну сайту.

**Швидкодія.** Мінусом фреймворку є швидкодія. Кількість запитів що він робить до бази за велика. Окрім того сам життєвий цикл, починаючи від диспетчеризації (розбивкою URL запиту) і закінчуючи відправкою зібраної сторінки до браузеру дуже повільна.

**Розмір дистрибутиву.** Розмір дистрибутиву сердній. CakePHP поставляється в базовій конфігурації без плагінів. Кількість базових класів не надто велика. Код не компресований.

**Зручність роботи.** Початково створювався як клон популярного Ruby on Rails, і багато ідей запозичено саме звідти. В ньому досить просто створювати прості речі, просто створити стандартну форму та провалідувати її, зберегти в базі, та при необхідності вивести повідомлення про помилку. Можна навіть змусити повністю згенерувати сторінку і код обробки даних згідно з існуючою таблицею за допомогою code folding'a. CakePHP'й Code folding можна навчити будувати сторінки згідно певного шаблону, втім CakePHP не змусити змінити логіку коду що генерується при code folding'у.

Та якщо треба зробити щось нестандартне, скажімо виводити текст помилки не знизу поля, а справа, то така здавалося б незначна задача може коштувати декількох годин роботи програміста. Та те що найбільше бісить при роботі с кейом, то є його непередбачуваність. Не завжди розумієш що він робить і чому саме так. Скажімо, ви намагаєтесь зробити вибірку з таблиці що зв'язана з іншою таблицею. Вказуєте параметри вибірки через "WHERE" а він може ці поля приписати не до таблиці з якої вибираємо а до зв'язаної. Не всі параметри функцій документовано, можливо через частоту виходу нових версій. Одним словом, без гарного знання фреймворку можна досить сильно затормозити розробку проекту. Проте в руках людей, що розібрались в ньому через читання коду самого фреймворку і знають всі його тонкощі CakePHP є потужною зброєю. Про CakePHP можна сказати що він використовує магію – про дуже багато речей він «здогадується» з назв таблиць, класів, функцій, їх порядку(і слід сказати що деякі речі зовсім неочевидні).

## **2. Фреймворк CodeIgniter.**

**Швидкодія.** Це дуже легковагий фреймворк і дуже швидкий. Швидкодія, практично як у чистого PHP. На його основі побудовано ще декілька фреймворків.

**Розмір дистрибутиву.** Найлегший з фреймворків. Вся логіка роботи фреймворку вміщується в один файл. Важко зрозуміти як він працює, проте зробити код компактнішим важко.

**Зручність роботи.** Перевагами CodeIgniter'a є його простота, швидкодія, він не робить нічого неочікуваного, все як написано в коді. Недоліки – досить скудні «хелпери» та функціональність. Так він не може сам ложити данні, що приходять \$\_POST в базу і сам не підготує данні для відображення, та ви хоч розумієте що відбувається! Одним словом не має «магії» в ньому, це можна вважати як плюсом так і мінусом. А ще він не буде сам запити до БД, як це робить CakePHP, згідно з прописаними зв'язками між таблицями в моделях.

## **3. Фреймворк Symfony.**

Багато хто вважає його «default framework». Дуже розповсюджений, гарно документований фреймворк. Має великі ком'юніті в тому числі російськомовне.

**Швидкодія.** Швидкодія нарікань не викликає. Можливість повністю кешувати сторінки значено підвищує продуктивність роботи системи. Для випадків коли кешування може спотворювати данні, можливе часткове кешування сторінки, шляхом відключення кешування вказаних елементів.

**Розмір дистрибутиву.** Значний. Велика кількість файлів. Близько 300х плагінів поставляються разом з Symfony.

**Зручність роботи.** Має розширені засоби фолдингу коду. Використовує Propel чи Doctrine ORM. ORM це об'єктно-реляційна проекція — технологія програмування, яка зв'язує бази даних з концепціями об'єктно-орієнтованих мов програмування, створюючи «віртуальну об'єктну базу даних». Практично тільки по «yaml» схемі для ORM може збудувати весь проект. Проте фронтенд частину все ж краще писати вручну. А от що

стосується адмін. частини, тут можна сміливо 70% генерувати через Symfony. Для Symfony дуже багато плагінів, що розширюють її функціональність. Робота коду досить передбачувана, проте інколи трапляються непорозуміння. Дуже багато чого є з коробки – зручна інтернаціоналізація, захист частин сайту від несанкціонованого доступу, кешування. Особливо хочеться відмітити, що сайт розділяється на 2 частини: для розробника (тут є спеціальне вікно відлагодження) і production версія.

#### 4. Порівняння коду.

Вибірка з БД.

```

Код CakePHP
    $allCategories = $this->Category->find('threaded');
Код CodeIgniter
    $query = $this->db->get('entries', 10);
    return $query->result();

```

```

Код Symfony
    $q = Doctrine_Query::create()
        ->from('JobeetJob j')
        ->where('j.expires_at > ?', date('Y-m-d H:i:s', time()));
    $this->jobeet_jobs = $q->execute();

```

Валідація даних

```

Код CakePHP
    var $validate = array(
        'title' => array(
            'rule' => 'notEmpty'
        ),
        'body' => array(
            'rule' => 'notEmpty'
        )
    );

```

```

Код CodeIgniter
    $this->load->library('form_validation');
    $this->form_validation->set_rules('username', 'Username',
    'callback_username_check');
    $this->form_validation->set_rules('password', 'Password',
    'required');
    $this->form_validation->set_rules('passconf', 'Password
    Confirmation', 'required');
    $this->form_validation->set_rules('email', 'Email', 'required');
    if ($this->form_validation->run() == FALSE)
    {$this->load->view('myform');}
    else
    {$this->load->view('formsuccess');}

```

```

Код Symfony
    $this->validatorSchema['email'] = new sfValidatorAnd(array(
        $this->validatorSchema['email'],
        new sfValidatorEmail(),
    ));

```

Контроль кешу

```

Код CakePHP
    var $helpers = array('Cache');
    var $cacheAction = array(
        'view' => array('callbacks' => true, 'duration' => 21600),
        'add' => array('callbacks' => true, 'duration' => 36000),
    );

```

```
'index' => array('callbacks' => true, 'duration' =>
48000));
```

*Код CodeIgniter*

```
$this->output->cache(n);
```

*Код Symfony*

```
default:
enabled:      false
with_layout:  false
lifetime:    86400
```

### 5. Порівняльна таблиця характеристик фреймворків

Фреймворк	Фолдинг коду	Наявність layout шаблону	Наявність Active Record для роботи з БД	Швидкодія	Кешування	Інтернаціо-налізація
CakePHP	+	+	+	низька	+	+
CodeIgniter	-	-	+	висока	-	+
Symfony	+	+	+	середня	+	+

**Висновки.** При виборі фреймворку необхідно користуватись слідуючи ми настановами. За допомогою фреймворк CodeIgniter можна дуже швидко написати невеликий проект на два – три дні, CakePHP – дозволить швидко створити проект, але оптимізувати його буде складно, Symfony – ідеально підходить під середні та великі проекти.

### Список літератури

1. Кузнецов М.В. / Самоучитель PHP 5/6; ВHV-Петербург, 2008. – 672 с.
2. Кухарчик А. / PHP: обучение на примерах; Новое знание; 2004. -240 с.
3. “Doctrine Documentation” / Doctrine community: [http://www.doctrine project. org/documentation/manual/1\\_2/en/introduction](http://www.doctrine project. org/documentation/manual/1_2/en/introduction).
4. “Cookbook, the CakePHP documentation” / Cake Software Foundation, Inc. <http://book.cakephp.org/>
5. “CodeIgniter User Guide” / Ellislab Inc. [http://codeigniter.com/user\\_guide/](http://codeigniter.com/user_guide/)
6. "Practical symfony" / Fabien Potencier : [http://www.symfony-project.org/jobeeet/1\\_4/Doctrine/ru/](http://www.symfony-project.org/jobeeet/1_4/Doctrine/ru/).

Одержано 16.08.10