

*Посвящается русскоязычному сообществу Yii.
Вы лучшие!*



ОГЛАВЛЕНИЕ

Предисловие	10
Рецензенты английской версии	11
Благодарности.....	12
Об авторе	13
Введение	14
Что вы найдёте в данной книге	14
Что понадобится для чтения этой книги	16
На кого рассчитана эта книга.....	17
Соглашения.....	17
Обратная связь.....	17
Глава 1. Под капотом.....	18
Вступление.....	18
Использование getters и setters	18
Использование событий Yii	21
Использование импорта и автозагрузки	29
Использование исключений	32
Настройка компонентов	35
Настройка виджетов по умолчанию	38
Использование коллекций ядра Yii	40
Работа с запросами.....	44
Глава 2. Маршрутизация, контроллеры и представления	48
Введение.....	48
Правила маршрутизации	49
Автоматическая генерация URL-адресов.....	52
Регулярные выражения в правилах маршрутизации	56
Правила маршрутизации для статических страниц.....	60
Добавление правил маршрутизации в рабочее приложение....	62
Базовый контроллер.....	66

Подключение внешних действий	68
Отображение статических страниц при помощи CViewAction.....	71
Использование flash-сообщений	73
Контекст контроллера в представлении.....	74
Повторное использование вложенных представлений.....	76
Клипы.....	78
Декораторы.....	80
Несколько макетов в приложении.....	81
Постраничная разбивка и сортировка данных.....	84
Глава 3. AJAX и jQuery.....	86
Введение.....	86
Загрузка блока через AJAX	86
Управление ресурсами	91
Подключение ресурсов.....	96
Работа с JSON	99
Передача параметров из PHP в JavaScript	102
Обработка переменного числа полей в форме	104
Глава 4. Работа с формами.....	111
Введение.....	111
Пишем свой валидатор.....	111
Загрузка файлов	114
Добавление CAPTCHA.....	118
Настройка CAPTCHA	122
Создаем виджет для ввода при помощи CInputWidget.....	125
Глава 5. Тестирование приложений.....	129
Введение.....	129
Настройка тестового окружения	129
Написание и запуск юнит-тестов	133
Фикстуры	139
Функциональное тестирование	144
Генерация отчетов о покрытии кода	149
Глава 6. База данных, Active record и трюки с моделями	153
Введение.....	153
Получение данных из базы данных	154
Создание и использование нескольких подключений к базам данных	160
Использование именованных групп условий для создания многоязычных моделей.....	163

Обработка полей модели с помощью методов-событий	
Active Record	167
Применение markdown и HTML.....	169
Подсветка кода с помощью Yii.....	172
Автоматический timestamp	178
Автоматическое указание автора.....	180
Реализация наследования с одной таблицей.....	182
Использование CDbCriteria.....	186
Глава 7. Использование компонентов Zii	188
Введение.....	188
Использование источников данных	188
Использование гридов	195
Использование списков.....	202
Создание своих столбцов грида	206
Глава 8. Расширение Yii	213
Введение.....	213
Создание поведений модели.....	213
Создание компонентов.....	220
Создание действий контроллера, пригодных для повторного использования	224
Создание контроллеров, пригодных для повторного использования	227
Создание виджета	231
Создание консольных команд.....	234
Создание фильтров	237
Создание модулей.....	239
Создание своего обработчика шаблонов	246
Подготовка расширений к публикации	251
Глава 9. Обработка ошибок, отладка и журналирование	255
Введение.....	255
Использование различных маршрутов для журналов	255
Анализ трассировки стека при ошибках.....	262
Журналирование и использование контекстной информации.....	265
Реализация собственного умного обработчика кода 404	270
Глава 10. Безопасность	275
Введение.....	275
Использование фильтров контроллера.....	275



Использование CHtml и CHtmlPurifier для предотвращения XSS.....	280
Предотвращение SQL-инъекций.....	285
Предотвращение CSRF.....	290
Использование RBAC	294
Глава 11. Настройка производительности.....	302
Введение.....	302
Использование передового опыта	302
Ускорение управления сессиями	308
Использование зависимостей кеша и цепочек	312
Профилирование приложений с помощью Yii	318
Глава 12. Использование постороннего кода ...	329
Введение.....	329
Использование Zend Framework из Yii	329
Настройка автозагрузчика Yii	334
Использование Kohana внутри Yii	339
Использование PEAR внутри Yii	346
Глава 13. Развёртывание	349
Введение.....	349
Изменение структуры директорий Yii	349
Перемещение приложения из корневой директории сервера	352
Совместное использование директории фреймворка	355
Перемещение части настроек в отдельные файлы.....	356
Использование нескольких конфигураций для упрощения развертывания	362
Реализация и исполнение заданий cron.....	366
Режим обслуживания	368



ПРЕДИСЛОВИЕ

Когда Александр рассказал мне, что собирается писать сборник рецептов по Yii, я задумался, насколько уникальными они будут, ведь в то время уже была создана официальная wiki, пополняющаяся силами сообщества. Мои опасения были напрасны.

Книга получилась полной информации об эффективном использовании фреймворка. Информация была подана настолько методично, что вполне могла использоваться как необходимое дополнение к полному руководству по Yii.

В процессе написания Александр просил членов команды Yii высказать своё мнение о черновиках и в процессе сумел заинтересовать всех нас. Как автор и ведущий разработчик Yii я считаю, что эту книгу должен прочитать каждый, кто работает с фреймворком.

В книге нет формального описания правил разработки. Вместо этого она показывает, как программировать на Yii с практической точки зрения. Материал особенно пригодится тем, кто работает со сжатыми сроками, так как в нём представлено множество решений проблем, с которыми разработчики могут столкнуться в своих проектах. Тем, кто уже знаком с фреймворком, книга также будет интересна. Большинство решений, представленных в данной книге, можно считать рекомендуемыми официально, так как они прошли тщательное рецензирование каждым членом команды Yii.

Александр показал себя данной книгой и его активным участием в проекте Yii как отличный программист и писатель.

*Qiang Xue,
ведущий разработчик фреймворка Yii*



РЕЦЕНЗЕНТЫ АНГЛИЙСКОЙ ВЕРСИИ

Anatoliy Dimitrov – обладатель сертификата O'Reilly по программированию PHP/MySQL и активный член сообщества Yii. Кроме того, он опытен в вопросах безопасности веб-приложений.

В прошлом занимал ведущие технические должности в нескольких крупнейших хостинговых компаниях и платёжных системах. Имеет за плечами множество успешных фриланс-проектов.

Antonio Ramirez Cobos (tonydspariard) – программист-самоучка. Нырнул в мир программирвоания в процессе изучения «железа» в TAFE в Мельбурне.

До встречи с PHP и чудесами OpenSource успел накопить более тринадцати лет опыта с Javascript, C++, Java, ASP.net (C#), Visual Basic (COM, COM+) и Dynamic DLL. После знакомства с PHP работал исключительно с ним и специализировался на веб-приложениях.

Любитель Yii, ведёт блог по адресу www.ramirezcobos.com. Активный участник форума Yii.

SAKURAI, atsushi – эксперт по микропроцессорам и PHP программист уже более десяти лет. В качестве руководителя команды разработчиков микропроцессоров занимался в том числе и сайтом для их поддержки. Благодаря Yii эффективность разработки в этом направлении заметно повысилась. Его главный вклад в Yii – перевод документации на японский язык.



БЛАГОДАРНОСТИ

Хочу поблагодарить команду разработчиков Yii, в особенности Qiang Xue, Maurizio Domba и Sebastian Thierer, которые рецензировали черновики данной книги и поддерживают Yii в отличной форме, постоянно его улучшая.

Спасибо Александру Кочетову, Antonio Ramirez Cobos и всем, кто присыпал замечания как по предварительной версии, так и по первой редакции. Ваши замечания очень помогли улучшить книгу.

Спасибо Qiang Xue и Wei Zhuo за Yii.

Спасибо Packt Publishing за предложение написать книгу и за то, что помогли её закончить.

Огромное спасибо тем, кто помог с русскоязычной версией: Александру Бордун, Ивану Левчунец, Сергею Сыцевич, Максиму Фуртуне и Дмитрию Мовчан.

Без вас я бы не справился!



ОБ АВТОРЕ

Александр Макаров – опытный инженер из Воронежа, успевший попробовать себя в роли РМ. Активный участник OpenSource проектов и один из разработчиков PHP-фреймворка Yii.

С 2008 по 2010 годы способствовал росту русскоязычного сообщества CodeIgniter. Примерно в то же время начал активно участвовать в OpenSource.

В 2009 году заинтересовался Yii, создал uiiframework.ru и закончил перевод официальной документации на русский язык.

С мая 2010 года присоединился к команде разработчиков фреймворка. Ведёт блог gmccreative.ru.

Выступает на различных конференциях. Работает в Stay.com, где занимается созданием крутых штук на Yii и не только.

Увлекается английским и русским языками, дизайном и UX. Любит смотреть хорошее кино, путешествовать и фотографировать.



ВВЕДЕНИЕ

Yii – очень гибкий и высокопроизводительный PHP-фреймворк, предназначенный для разработки веб-приложений. От небольших страничек до масштабируемых приложений уровня предприятия. Название фреймворка расшифровывается как Yes It Is, что является точным ответом на большинство вопросов, пока ещё не знакомых с Yii, разработчиков: «А он быстр?», «Безопасен?», «Подходит ли для профессиональной разработки?», «Подойдёт ли для моего следующего проекта?». Ответ на все эти вопросы один: «Yes, it is!»

Данный сборник рецептов состоит из 13 независимых глав, каждая из которых полна решений, которые помогут вам эффективно использовать фреймворк. Вы узнаете о скрытых возможностях, работе ядра, создании своих компонент, разработке через тестирование и ещё множестве интересных тем.

Что вы найдёте в данной книге

Глава 1. Под капотом

Рассказывает о наиболее интересных возможностях Yii, про которые мало говорится в официальном руководстве: события, импорт и автозагрузка классов, исключения, компоненты, настройка виджетов и другое.

Глава 2. Маршрутизация, контроллеры и представления

О полезных приёмах, относящихся к обработке и построению URL, контроллерам и представлениям. В данной главе описаны правила URL, внешние действия контроллера и сами контроллеры, декораторы представлений и многое другое.

Глава 3. AJAX и jQuery

Поведает о клиентской части Yii, в которой используется jQuery – самая широко применяемая JavaScript-библиотека в мире. Главным образом рассматриваются особенности Yii, а не самой jQuery.

Глава 4. Работа с формами

Yii сильно облегчает работу с формами. Несмотря на то, что документация на эту тему довольно полная, есть некоторые особенности, требующие разъяснений и примеров.

В этой главе среди прочего описывается создание своих валидаторов и виджетов форм, загрузка файлов, использование и настройка CAPTCHA.

Глава 5. Тестирование приложений

Описывает, помимо модульного тестирования, функциональное тестирование и генерацию отчёта по покрытию кода тестами. В рецептах этой главы применяется разработка через тестирование: сначала пишутся тесты, после разрабатывается само приложение.

Глава 6. База данных, Active record и трюки с моделями

Показывает, как эффективно работать с базой данных как при использовании моделей, так и напрямую, через DAO. Рассказывает про работу с несколькими базами данных одновременно, про построение и использование критерия запроса и предварительную обработку полей модели.

Глава 7. Использование компонентов Zii

Посвящена провайдерам данных, гридам и спискам. Учит настраивать сортировку и поиск, использовать грид с несколькими связанными моделями, создавать свои типы столбцов и многому другому.

Глава 8. Расширение Yii

Не только учит реализации своих расширений Yii, но и рассказывает, как сделать их действительно гибкими и полезными для сообщества. В дополнение мы рассмотрим множество деталей, позволяющих сделать ваше расширение лучше.

Глава 9. Обработка ошибок, отладка и журналирование

Журналирование, анализ стека исключения, реализация своего обработчика ошибок.

Глава 10. Безопасность

Необходимая информация о том, как сделать приложение безопасным в соответствии с главным принципом «фильтруй входные данные, экранируй выходные». Рассматриваются такие темы, как создание своих фильтров контроллера, предотвращение атак типа XSS, CSRF и SQL-инъекций. Экранирование выходных данных и использование контроля доступа на основе ролей.

Глава 11. Настройка производительности

Показывает как настроить Yii для получения повышенной производительности. В этой главе вы познакомитесь с приёмами разработки, которые позволят вашему приложению работать достаточно быстро для того, чтобы дорасти до высоких нагрузок.

Глава 12. Использование стороннего кода

Использование стороннего кода и библиотек в приложениях на Yii. Рассматриваются Zend Framework, Kohana и PEAR, но после того, как поймёте, как это работает, вы сможете использовать любые другие библиотеки.

Глава 13. Развёртывание

Данная глава познакомит вас с различными приёмами, которые особенно полезны при развёртывании приложения, работе в команде и позволяют сделать жизнь разработчика проще и удобней.

Что понадобится для чтения этой книги

Для того, чтобы запускать примеры из этой книги, потребуется следующее:

- Apache 2.x. Другие вебсерверы тоже будут работать, но инструкции приведены именно для Apache.
- MySQL 5 с поддержкой InnoDB.
- PHP любой версии, начиная с 5.2.
- Последняя версия Yii 1.1.x.

Для некоторых рецептов дополнительно потребуется следующее:

- PHPUnit;
- XDebug;
- Selenium RC;
- PEAR;
- Smarty;
- memcached.

На кого рассчитана эта книга

Если вы разработчик с хорошим знанием PHP5, знаете основы Yii, хотя бы бегло прочли полное руководство и уже попробовали разрабатывать свои приложения, можете смело начинать читать эту книгу. Знание ООП и MVC пригодится, потому как в Yii они используются повсеместно.

Соглашения

В этой книге вы встретите некоторое количество стилей текста, которые помогают различать различные блоки информации.

Код, названия файлов и пути выделены
моноширинным шрифтом.

Консольный ввод и вывод также оформлены
моноширинным жирным шрифтом.

Те части, на которые стоит обратить внимание, выделены
жирным шрифтом.

Предупреждения и важные заметки выделены в отдельные блоки.

Обратная связь

Ваши вопросы, замечания и пожелания всегда кстати. Связаться с автором можно через специальную форму по адресу <http://ru.yiicookbook.org/feedback>. Через эту же форму можно отправить найденные в книге ошибки. Они будут размещены по адресу <http://ru.yiicookbook.org/errata>.

Код к книге можно найти по адресу <http://ru.yiicookbook.org/code>.



ГЛАВА 1.

Под капотом

В этой главе мы рассмотрим:

- Использование getters и setters.
- Использование событий Yii.
- Использование импорта и автозагрузки.
- Использование исключений.
- Настройка компонентов.
- Настройка виджетов по умолчанию.
- Использование коллекций.
- Работа с запросами.

Вступление

В этой главе мы рассмотрим самые интересные особенности Yii, которые скрыты «под колпаком». Главным образом они описаны в API фреймворка, но поскольку они не упомянуты в официальном руководстве (<http://www.yiiframework.com/doc/guide/>) или же упомянуты очень поверхностно, обычно их используют только опытные разработчики. Тем не менее, особенности, описанные здесь, являются относительно простыми, и их использование делает разработку с Yii намного более увлекательной и продуктивной.

Использование getters и setters

В Yii присутствует много особенностей, которые пришли из других языков, таких как Java или C#. Одна из них – это определение свойств с помощью геттеров (getters) и сеттеров (setters) для любого класса, который наследует класс `CComponent` (то есть практически любого класса Yii).

Из этого рецепта вы научитесь определять свои собственные свойства, используя getters и setters делать свойства доступными только для

чтения и скрывать собственную обработку сохраняя обычный синтаксис присваивания значения PHP.

Как это делается

1. Так как в PHP свойства отсутствуют на уровне языка, мы можем использовать только методы получения (getters) и назначения (setters) следующим образом:

```
class MyClass
{
    // скрытие $property
    private $property;

    // получение
    public function getProperty()
    {
        return $this->property;
    }

    // назначение
    public function setProperty($value)
    {
        $this->property = $value;
    }
}

$object = new MyClass();

// установка значения
$object->setProperty('value');

// получение значения
echo $object->getProperty();
```

2. Этот синтаксис очень распространен в мире Java, но требует немного больше времени для использования в PHP. Тем не менее, мы хотим использовать ту же функциональность, которую нам дают свойства C#: вызов методов получения и назначения как членов класса. С Yii мы можем сделать это следующим образом:

```
// необходимо наследовать CComponent
class MyClass extends CComponent
{
    private $property;
    public function getProperty()
    {
```

```
        return $this->property;
    }
    public function setProperty($value)
    {
        $this->property = $value;
    }
}

$object = new MyClass();
$object->property = 'value'; // равнозначно $object
->setProperty('value');
echo $object->property; // равнозначно $object
->getProperty();
```

3. Используя эту функцию, вы можете сделать свойства только для чтения или только для записи, сохраняя при этом простой синтаксис PHP следующим образом:

```
class MyClass extends CComponent
{
    private $read = 'read only property';
    private $write = 'write only property';

    public function getRead()
    {
        return $this->read;
    }

    public function setWrite($value)
    {
        $this->write = $value;
    }
}

$object = new MyClass();

// дает ошибку, так как мы пытаемся изменить
// свойство, доступное только для чтения
$object->read = 'value';

// выводит 'read only property'
echo $object->read;

// дает ошибку, так как мы пытаемся получить
// свойство, доступное только для записи
echo $object->write;

// запишет 'value' в приватное поле $write
$object->write = 'value';
```

4. Yii широко использует эту технику, так как почти всё – это компоненты. Например, когда вы вызываете `Yii::app() -> user->id`, чтобы получить ID текущего авторизованного пользователя, то на самом деле вызывается `Yii::app() -> getUser() ->getId()`.

Как это работает

Для использования геттеров и сеттеров как свойств `CComponent` использует магические методы PHP: `__get`, `__set`, `__isset` и `__unset` (<http://php.net/manual/en/language.oop5.magic.php>). Следующий пример показывает, как в Yii выглядит `CComponent::__get`:

```
public function __get($name)
{
    $getter='get' . $name;
    if(method_exists($this,$getter))
        return $this->$getter();
    ...
}
```

Магический метод PHP перехватывает все вызовы к недостающим реальным свойствам, поэтому, когда мы вызываем `$myClass->property`, он получает свойство как параметр `$name`. Если же существует метод с именем `getProperty`, тогда PHP использует его возвращаемое значение как значение свойства.

И ещё

Для получения дополнительной информации обратитесь по следующей ссылке: <http://www.php.net/manual/en/language.oop5.overloading.php#language.oop5.overloading.members>.

Смотрите также

- Рецепт «Использование событий Yii» в этой главе.
- Рецепт «Настройка компонентов» в этой главе.

Использование событий Yii

Большинство классов фреймворка наследуют `CComponent`, что позволяет достичь большой гибкости приложений с помощью событий. Событие представляет собой сообщение о том, что приложение что-то выполнило. Мы можем зарегистрировать несколько обработчиков, которые будут реагировать на определенные типы событий. Обра-

ботчик может получить параметры от события, с которым работает, и реагировать соответствующим образом.

В этом рецепте вы узнаете, как объявлять и использовать как предопределенные, так и пользовательские события.

Как это делается

Чтобы объявить событие в дочернем классе `CComponent`, вы должны добавить метод с именем, начинающимся с `on`. Например, если вы добавите метод `onRegister`, то получите соответствующее объявленное событие.

Метод, который объявляет событие, является обработчиком события по умолчанию.

Обычно события используются следующим образом:

- объявите событие, добавив соответствующий метод;
- подключите один или несколько обработчиков событий;
- компонент вызывает событие с помощью метода `CComponent::raiseEvent`;
- все подписанные обработчики автоматически вызываются.

Давайте посмотрим, как мы можем присоединить обработчик к событию. Чтобы сделать это, мы можем использовать метод `CComponent::attachEventHandler`. Он принимает два параметра:

- `$name`: имя события;
- `$handler`: обработчик события; стандартный обратный вызов PHP.

В PHP у нас есть несколько способов определения обратного вызова:

- использовать глобальную функцию и просто передать ее в виде строки, как, например, `'my_function'`;
- использовать статический метод класса: `array('ClassName', 'staticMethodName')`;
- использовать метод объекта: `array($object, 'objectMethod')`;
- создать и передать анонимную функцию с помощью `create_function`, например:

```
$component->attachEventHandler('onClick',
    create_function('$event', 'echo "Click!";'));
```

- начиная с PHP 5.3, вы можете использовать анонимные функции без `create_function`:

```
$component->attachEventHandler('onClick',
    function($event){ echo "Click!"; })
```

При использовании `CComponent::attachEventHandler` обработчик события добавляется в конец списка обработчиков.

- Чтобы сократить объём кода, вы можете использовать свойства компонента для управления обработчиками событий следующим образом:

```
$component->onClick=$handler;
// или:
$component->onClick->add($handler);
```

- Для более точного управления обработчиками событий вы можете получить список обработчиков (`cList`) с помощью `CComponent::getEventHandlers` и работать с ним. Например, вы можете присоединить обработчик события так же, как с `attachEventHandler`, используя следующий код:

```
$component->getEventHandlers('onClick')->add($handler);
```

- Чтобы добавить обработчик события в начало списка обработчиков, используйте:

```
$component->getEventHandlers('onClick')->insertAt
(0, $handler);
```

- Чтобы удалить определенный обработчик, можно использовать `CComponent::detachEventHandler` следующим образом:
- ```
$component->detachEventHandler('onClick', $handler);
```
- Кроме этого, можете получить список обработчиков, как было показано ранее, удалять обработчики из него.

`CComponent::hasEvent` проверяет, определено ли указанное событие в компоненте. `CComponent::hasEventHandler` проверяет, есть ли присоединенные к указанному событию.

Поскольку мы теперь знаем, как определить и использовать обработчики, давайте рассмотрим некоторые примеры из реальной жизни:

- для ускорения загрузки страницы принято сжимать данные при помощи `gzip` непосредственно перед отдачей. Если у вас есть доступ к тонкой настройке сервера, то вы можете этим воспользоваться, однако в некоторых средах, таких как виртуальный хостинг, вам это не удастся;

Конец ознакомительного фрагмента.

Приобрести книгу можно  
в интернет-магазине  
«Электронный универс»  
[e-Univers.ru](http://e-Univers.ru)