Компоненты. Фреймворки.

- **CKEditor**
- **KCAPTCHA**
- **Обзор фреймворков**
- **Yii**

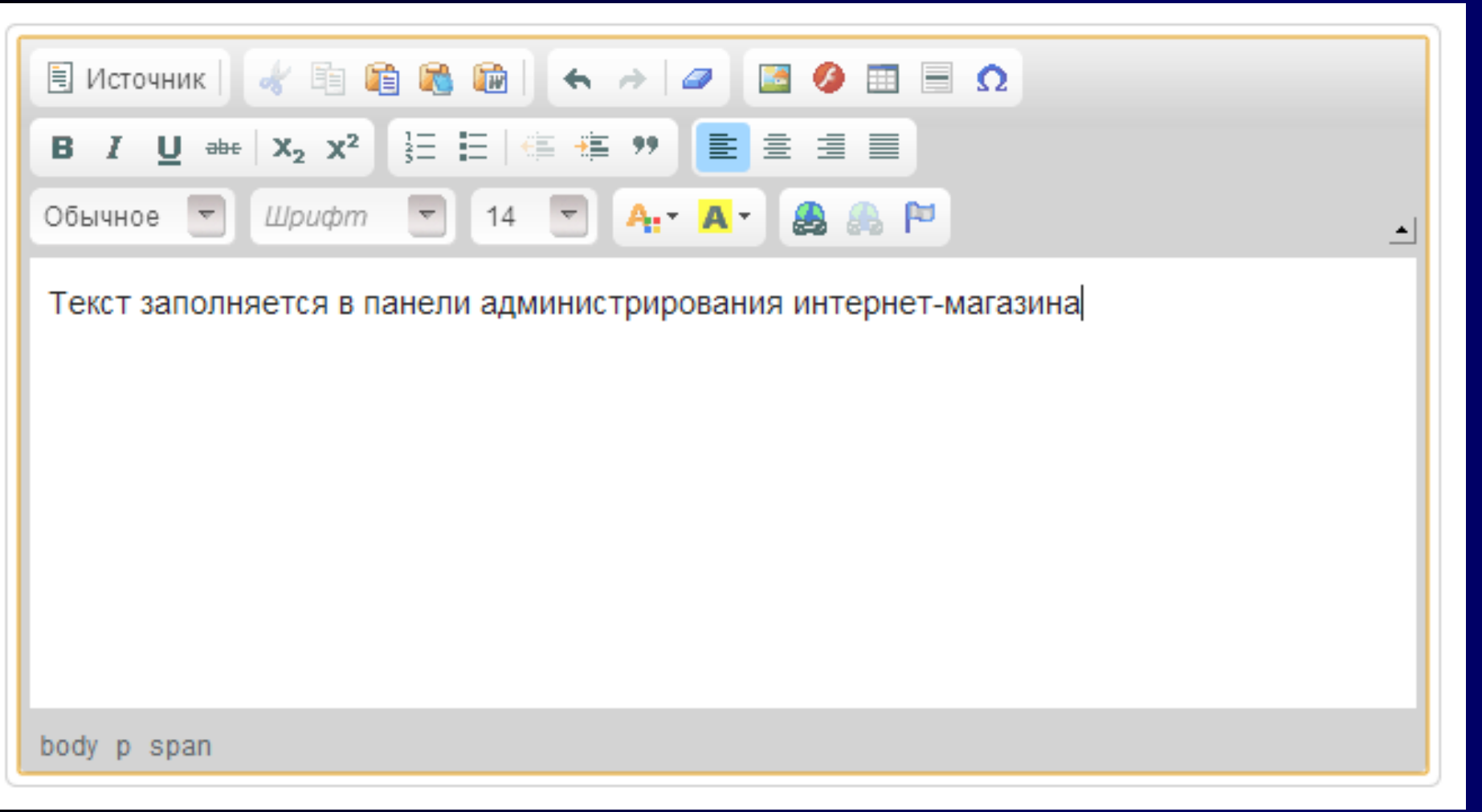
Компоненты.

- При разработке web-сайтов часто требуются типовые решения, такие, как online-редактор HTML-кода, автоматизированный тест Тьюринга для защиты от спама в досках объявлений, и многие другие. Подобные модули не сложно разработать самостоятельно, однако существуют готовые решения с открытым исходным кодом. Рассмотрим подключение и применение двух наиболее популярных решений.

Компоненты.

- Текстовый редактор CKeditor обладает большим количеством разнообразных функций и поддерживается большинством современных браузеров. Он оформлен в стиле MS Word, что делает обращение с ним более привычным.
- Его можно скачать с сайта разработчика http://ckeditor.com. Существует его реализации почти для десятка языков программирования, но рассмотрим только на php. Распакованный архив копируется в папку сайта, после чего необходимо подключить необходимый файл.

Компоненты.


Источник
Обычное
Шрифт
14
Текст заполняется в панели администрирования интернет-магазина
body p span

Компоненты.

- Подключим необходимый файл:
- `<?php`
- `include_once("fckeditor/fckeditor.php") ;`
- `?>`
- Для использования редактора добавляем следующий код внутри <FORM>:
- `<form action="sampleposteddata.php" method="post" target="_blank">`
- `<?php $oFCKeditor = new FCKeditor('FCKeditor1') ;`

Компоненты.

- `$oFCKeditor->BasePath= '/fckeditor/' ;`
- `$oFCKeditor->Value = '<p>This is some <strong>sample text</strong>. You are using <a href="http://www.fckeditor.net/"> FCKeditor</a>.</p>';`
- `$oFCKeditor->Create();`
- `?>`
- `<br><input type="submit" value="Submit"></form>`

Компоненты.

- Если форма размещается на странице, вход на которую не требует авторизации, то гостевая книга или доска объявлений окажутся быстро переполненными спамом, рассылаемом соответствующими программами. Для того, чтобы разрешить добавление записей только человеком, требуется провести автоматизированный обратный тест Тьюринга, известный также, как CAPTCHA (от англ. Completely Automated Public Turing test to tell Computers and Humans Apart — полностью автоматизированный публичный тест Тьюринга для различения компьютеров и людей).

Компоненты.

- Иными словами, это задача, которую легко решает человек, но которую не может выполнить компьютер. Чаще всего CAPTCHA выглядит как тем или иным образом зашумленное случайное число, слово или иная надпись, которую пользователю нужно прочитать и ввести прочитанный результат, хотя существуют и другие алгоритмы.
- Проект KCAPTCHA — это готовое решение, написанное на языке PHP, которое вы можете бесплатно скачать и установить на свой сайт для защиты от спама и флуда.

Компоненты.

- Проект KCAPTCHA ставит перед собой цель предложить программисту решение с одной стороны весьма защищенное, с другой — максимально малотребовательное к ресурсам и конфигурации хостинга. Пример:
- `<img src="kcaptcha/code.php?<?php echo session_name()?>=<?php echo session_id()?>"></p>`

Компоненты.

- Принцип действия: скрипт стартует сессию и записывает в нее под именем $_SESSION['captcha_keystring'] случайным образом сгенерированную строку, после чего выдает изображение, содержащее эту самую строку в зашумленном виде. При проверке пользовательского ввода остается только прочитать из сессии кодовую строку и сравнить с тем, что ввел пользователь. Системные требования: PHP версии 4.0.6 и выше с поддержкой GD версии 2.

Компоненты.

- В комплект входит набор растровых шрифтов, так что скрипт, скорее всего, будет сразу готов к работе, не требуя установки дополнительных компонент.
- Можно настраивать цвета и набор символов, применяемые при создании изображения. Все настройки, касающиеся KCAPTCHA, располагаются в файле kcaptcha_config.php. Здесь можно задать вид отображаемого проверочного кода.
- Скачать KCAPTCHA 2.0 можно здесь: http://www.captcha.ru/kcaptcha.zip

Фреймворки PHP

- **Фреймворк** (англ. *framework* — каркас, структура) — структура программной системы; программное обеспечение, облегчающее разработку и объединение разных компонентов большого программного проекта. Подход к построению программ, где любая конфигурация программы строится из двух частей: первая, постоянная часть — каркас, не меняющийся от конфигурации к конфигурации и несущий в себе гнезда, в которых размещается вторая, переменная часть — сменные модули (или точки расширения).

Фреймворки PHP

- Фреймворк отличается от понятия библиотеки тем, что библиотека может быть использована в программном продукте просто как набор подпрограмм близкой функциональности, не влияя на архитектуру программного продукта и не накладывая на неё никаких ограничений. В то время как фреймворк диктует правила построения архитектуры приложения, задавая на начальном этапе разработки поведение по умолчанию, каркас, который нужно будет расширять и изменять согласно указанным требованиям.

Фреймворки PHP

- Также, в отличие от библиотеки, которая объединяет в себе набор близкой функциональности, фреймворк может содержать в себе большое число разных по тематике библиотек. Фреймворк определяется как множество конкретных и абстрактных классов, а также определений способов их взаимоотношения. Конкретные классы обычно реализуют взаимные отношения между классами. Абстрактные классы представляют собой *точки расширения*, в которых каркасы могут быть использованы или адаптированы.
- .

Фреймворки PHP

- *Точка расширения* — это та часть фреймворка, для которой не приведена реализация.
- Идея фреймворка - это предложить движок, который можно использовать для нескольких приложений. Все приложения имеют ряд основных общих черт - какой-то интерфейс с базой данных, некоторое количество логики приложения, то, что необходимо для пользователя. Фреймворк разработан таким образом, чтобы:
  - обеспечить структуру всех общих элементов (взаимодействия с базой данных, уровень представления, логика приложения),

Фреймворки PHP

- тратить меньше времени на написание кода базы данных интерфейса или представления интерфейса, и это даёт больше времени на написание самого приложения.

- Архитектура которая сломала привычное представление, называется Model-View-Controller (MVC). Model ссылается на данные, View - на уровень представления, а Controller ссылается на приложение или бизнес-логику.

Фреймворки PHP

- **Model-view-controller** (**MVC**, «модель-представление-поведение», «модель-представление-контроллер», «модель-вид-контроллер») — схема использования нескольких шаблонов проектирования, с помощью которых модель данных приложения, пользовательский интерфейс и взаимодействие с пользователем разделены на три отдельных компонента. Таким образом, чтобы модификация одного из компонентов оказывала минимальное воздействие на остальные.

Фреймворки PHP

- Концепция MVC была описана в 1979 году Трюгве Реенскауг, тогда работающим над языком программирования Smalltalk в Xerox. Оригинальная реализация описана в статье «Applications Programming in Smalltalk-80: How to use Model-View-Controller». В оригинальной концепции была описана сама идея и роль каждого из элементов: модели, представления и контроллера. Но связи между ними были описаны без конкретизации.

Фреймворки PHP

- **Назначение**
- Основная цель применения этой концепции состоит в разделении бизнес-логики (*модели*) от её визуализации (*представления, вида*). За счет такого разделения повышается возможность повторного использования. Наиболее полезно применение данной концепции в тех случаях, когда пользователь должен видеть те же самые данные одновременно в различных контекстах и/или с различных точек зрения. В частности, выполняются следующие задачи:

Фреймворки PHP

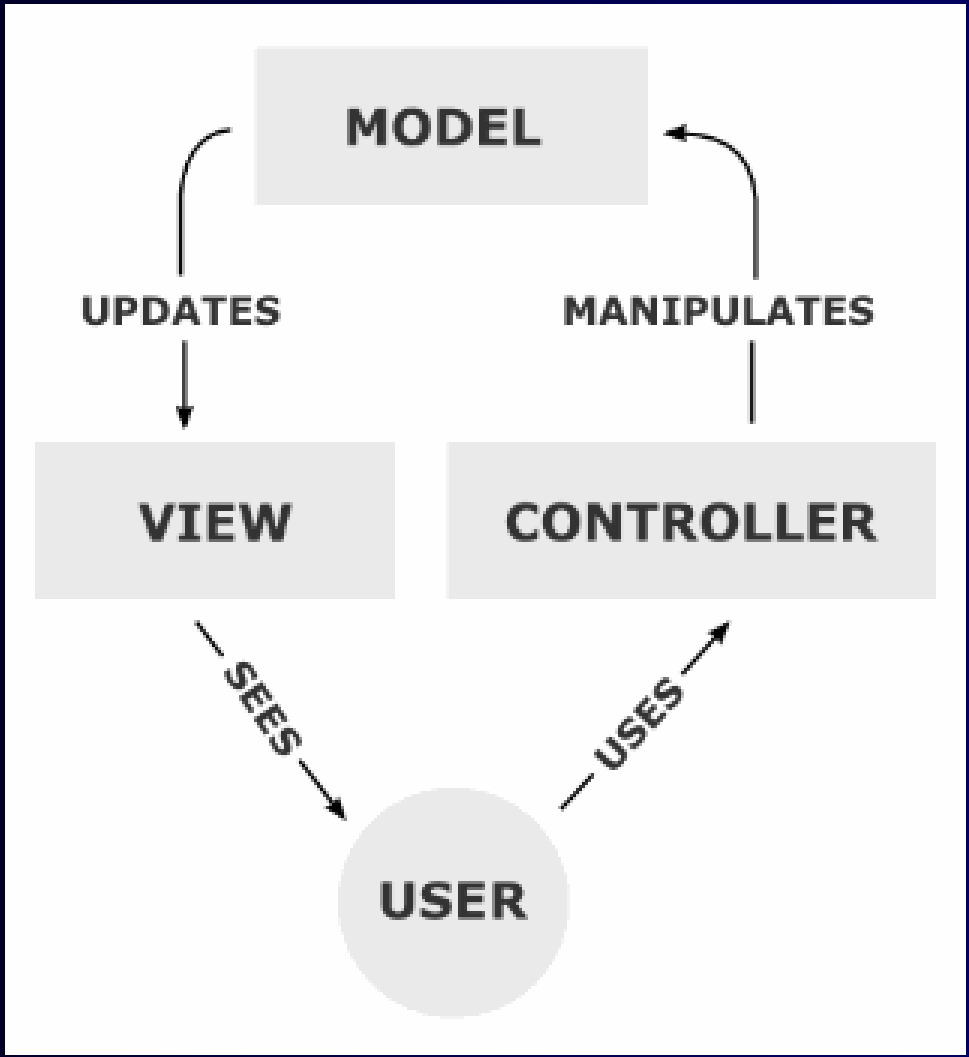
1. К одной *модели* можно присоединить несколько *видов*, при этом не затрагивая реализацию *модели*. Например, некоторые данные могут быть одновременно представлены в виде электронной таблицы, гистограммы и круговой диаграммы.
2. Не затрагивая реализацию *видов*, можно изменить реакции на действия пользователя (нажатие мышью на кнопке, ввод данных), для этого достаточно использовать другой *контроллер*.

Фреймворки PHP

3. Ряд разработчиков специализируется только в одной из областей: либо разрабатывают графический интерфейс, либо разрабатывают бизнес-логику. Поэтому возможно добиться того, что программисты, занимающиеся разработкой бизнес-логики (*модели*), вообще не будут осведомлены о том, какое *представление* будет использоваться.

- **Концепция**
- Концепция MVC позволяет разделить данные, представление и обработку действий пользователя на три отдельных компонента:

Фреймворки PHP



Фреймворки PHP

- Модель. Модель предоставляет знания: данные и методы работы с этими данными, реагирует на запросы, изменяя своё состояние. Не содержит информации, как эти знания можно визуализировать.
- Представление, вид. Отвечает за отображение информации (визуализацию). Часто в качестве представления выступает форма (окно) с графическими элементами.
- Контроллер. Обеспечивает связь между пользователем и системой: контролирует ввод данных пользователем и использует модель и вид для реализации необходимой реакции.

Фреймворки PHP

- Как *представление*, так и *контроллер* зависят от *модели*. Однако *модель* не зависит ни от *представления*, ни от *контроллера*. Тем самым достигается назначение такого разделения: оно позволяет строить *модель* независимо от *визуального представления*, а также создавать несколько различных *представлений* для одной *модели*.
- Для реализации схемы MVC используется достаточно большое число шаблонов проектирования, основные из которых «наблюдатель», «стратегия», «компоновщик».

Фреймворки PHP

- Наиболее типичная реализация отделяет вид от модели путем установления между ними протокола взаимодействия, используя аппарат событий (подписка/оповещение). При каждом изменении внутренних данных в модели она оповещает все зависящие от неё представления, и представление обновляется. Для этого используется шаблон «наблюдатель». При обработке реакции пользователя вид выбирает, в зависимости от нужной реакции, нужный контроллер, который обеспечит ту или иную связь с моделью.

Фреймворки PHP

- Для этого используется шаблон «стратегия», или вместо этого может быть модификация с использованием шаблона «команда».
- А для возможности однотипного обращения с подобъектами сложно-составного иерархического вида может использоваться шаблон «компоновщик». Кроме того, могут использоваться и другие шаблоны проектирования, например, «фабричный метод», который позволит задать по умолчанию тип контроллера для соответствующего вида.

Фреймворки PHP

- Представим сравнение 6-ти популярных фреймворков по основным критериям (необходимый уровень знаний, сфера применения, документация и т.д.). Автор: Александр Макаров.
- **Zend Framework 1**. Академически грамотный код. Очень гибок. Требует хорошего знания PHP и ООП. Придётся немного доводить под себя прежде, чем использовать. Сухая, но достаточно полная техническая документация.
- **Необходимый уровень знаний:** PHP5, ООП, шаблоны проектирования.

Фреймворки PHP

- **Предполагаемые проекты:** Средние — большие.
- **Сложность установки и настройки:** Высокая.
- **Требует настройки:** Много.
- **Документация и примеры:** Хорошая.
- **Русскоязычное сообщество:** http://zendframework.ru.
- **Лицензия:** New BSD.
- **CakePHP.** Много встроенного функционала. Всё довольно тесно интегрировано. Документация не в лучшем состоянии.

Фреймворки PHP

- **Необходимый уровень знаний:** PHP, ООП, умение разбираться в исходном коде фреймворка.
- **Предполагаемые проекты:** Маленькие — средние.
- **Сложность установки и настройки:** Низкая.
- **Требует настройки:** Немного.
- **Документация и примеры:** Имеется.
- **Русскоязычное сообщество:** Почти не активно.
- **Лицензия:** MIT.

Фреймворки PHP

- **Code Igniter 2**. Почти микрофреймворк. Очень лёгок для изучения. Отличная документация. Гибок. Легко использовать сторонний код.
- **Необходимый уровень знаний:** PHP, Основы ООП.
- **Предполагаемые проекты:** Маленькие — большие.
- **Сложность установки и настройки:** Низкая.
- **Требует настройки:** Немного.
- **Документация и примеры:** Отличная.
- **Русскоязычное сообщество:** Документация, форум, блоги. **Лицензия:** Своя.

Фреймворки PHP

- **Kohana 3**. Быстр, гибок. Свой подход к модульности. Скудная документация.
- **Необходимый уровень знаний:** PHP5, ООП.
- **Предполагаемые проекты:** Маленькие — средние.
- **Сложность установки и настройки:** Низкая.
- **Требует настройки:** Немного.
- **Документация и примеры:** Скудная, местами отстаёт от кода.
- **Русскоязычное сообщество:** Нет.
- **Лицензия:** BSD-style.

Фреймворки PHP

- Symfony 2. Активно использует командную строку, yaml. Хорошая система view, генераторы кода, dependency injection для всего. Изучить очень непросто, несмотря на хорошую документацию.
- **Необходимый уровень знаний:** PHP5, ООП, ORM, консоль.
- **Предполагаемые проекты:** Большие.
- **Сложность установки и настройки:** Высокая.
- **Требует настройки:** Много.
- **Документация и примеры:** В процессе написания.

Фреймворки PHP

- **Русскоязычное сообщество:** Нет.
- **Лицензия:** MIT.
- **Yii 1.1**. Проще в изучении, чем Zend и Symfony. Хорошая система view, генераторы кода. Довольно тесная интеграция.
- **Необходимый уровень знаний:** PHP5, ООП.
- **Предполагаемые проекты:** Маленькие — большие.
- **Сложность установки и настройки:** Средняя.
- **Требует настройки:** Немного.
- **Документация и примеры:** Отличная.

Фреймворки PHP

- **Русскоязычное сообщество:** Документация, форум, блоги.
- **Лицензия:** New BSD.
- Рассмотрим особенности последнего фреймворка подробнее. **Yii** (акроним от «Yes It Is!», произносится как «Yee» или [ji:]) — веб-фреймворк, написанный на PHP, и реализующий парадигму MVC.
- История Yii началась 1 января 2008 года, как проект по исправлению изъянов в фреймворке PRADO (PHP Rapid Application Development Object-oriented.

Фреймворки PHP

- Фреймворк PRADO был попыткой перенести ASP.NET на платформу PHP. PRADO унаследовал от ASP.NET почти все отрицательные стороны: медленно обрабатывал сложные страницы, имел крутую кривую обучения и был довольно труден в настройке.
- В определенный момент автор (Qiang Xue) понял, что PHP-фреймворк должен быть построен несколько по-другому, и вот 3 декабря 2008 был выпущен Yii 1.0.

Фреймворки PHP

- **Возможности**
  - Высокая производительность относительно других фреймворков написанных на PHP
  - Парадигма Модель-вид-контроллер
  - Интерфейсы DAO и ActiveRecord для работы с базами данных (PDO)
  - Поддержка интернационализации
  - Кэширование страниц и отдельных фрагментов
  - Перехват и обработка ошибок
  - Ввод и валидация форм
  - Аутентификация и авторизация
  - Использование AJAX и интеграция с jQuery
  - Поддержка тем оформления для их лёгкой смены

Фреймворки PHP

- Возможность подключения сторонних библиотек
- Миграции базы данных
- Автоматическое тестирование
- Поддержка REST

- **REST** (сокр. англ. *Representational State Transfer*, «передача состояния представления» или «передача репрезентативного состояния») — стиль построения архитектуры распределенного приложения. Был описан и популяризован в 2000 году Роем Филдингом (Roy Fielding), одним из создателей протокола HTTP.

Фреймворки PHP

- Самой известной системой, построенной в значительной степени по архитектуре REST, является современная Всемирная паутина.
- Данные в REST должны передаваться в виде небольшого количества стандартных форматов (например HTML, XML, JSON). Сетевой протокол (как и HTTP) должен поддерживать кэширование, не должен зависеть от сетевого слоя, не должен сохранять информацию о состоянии между парами «запрос-ответ». Утверждается, что такой подход обеспечивает масштабируемость системы и позволяет ей эволюционировать с новыми требованиями.

Фреймворки PHP

- Антиподом REST является подход, основанный на вызове удаленных процедур (Remote Procedure Call — RPC). Подход RPC позволяет использовать небольшое количество сетевых ресурсов с большим количеством методов и сложным протоколом. При подходе REST количество методов и сложность протокола строго ограничены, из-за чего количество отдельных ресурсов может быть большим.

Фреймворки PHP

- В блогосфере можно найти массу статей со сравнительным анализом фреймворков. В целом, прослеживаются следующие тенденции:
  - Yii активно развивается.
  - Yii не выглядит «монстром» по сравнению с фреймворками symfony и Zend Framework (у которых число строк кода соизмеримо с числом строк кода операционных систем).
  - В некоторых сравнительных работах отмечается высокая скорость изучения фреймворка, получения результатов и прототипирования по сравнении с Zend_Framework и Symfony. Также отмечается его стабильность и безопасность.

Фреймворки PHP

- Текущая Версия: 1.1.14, Дата выпуска: 11 августа 2013, Окончание поддержки: 31 декабря 2015, Системные требования: PHP 5.1.0 или выше.
- Литература: Макаров А. Yii Сборник рецептов Более 80 рецептов… Изд-во: ДМК Пресс, 2013. - 372 с.
- http://yiiframework.ru/