

Как разрабатывается фреймворк Yii

Александр Макаров, Yii core team



<http://www.devconf.ru>

Что такое Yii?

- ✓ PHP5 MVC фреймворк.
- ✓ Лёгкий и быстрый.
- ✓ Мощный и гибкий.
- ✓ Красивый API.
- ✓ Пересекается с Prado, Rails, Symfony и... Joomla (в хорошем смысле).
- ✓ Свой ActiveRecord.
- ✓ i18n на основе CLDR.
- ✓ Кэш с зависимостями и кучей бэкэндов.
- ✓ RBAC.
- ✓ Консоль.
- ✓ Генератор кода.
- ✓ ...

Prado (2004) → Yii 1.0 (2008) → Yii 1.1 (2010) → Yii2 (?)

Типичный action

```
public function actionView($id)
{
    $post = Post::model()->findByPk($id);
    if(!$post)
        throw new CHttpException(404);

    $this->render('view', array(
        'post' => $post,
    ));
}
```

Более сложный Active Record

```
$posts = Post::model()->taggedWith(array('yii',  
'DevConf', 'PHP'))->published()->with('comments')->findAll();
```

```
foreach($posts as $post){  
    echo $post->title;  
    foreach($post->comments as $comment){  
        echo $comment->text;  
    }  
}
```

Откуда такое название?!

- Yes It Is (2010).
- Easy Efficient Extensible (2009).
- Китайский символ Yi:
 - Лёгкость и простота.
 - Изменчивость и гибкость.
 - Постоянство. Главные принципы не меняются.



“Yi ching”, “Книга перемен”.

Зачем велосипед и почему MVC?

- Во времена Prado не было нормальной альтернативы.
- Prado заимствовал у ASP.NET. Чужеродно.
- MVC популярен и удобен.
- Существующие в 2008-м году фреймворки не устраивали.

Иногда велосипед — это
хорошо!

Заимствование

- Стоит ли смотреть на других?
- Бездумное копирование – дело гиблое.
- Анализ необходим.



Кто «владелец» фреймворка?

Сообщество

- ✓ Фреймворк не подпиливается под конкретные проекты.
- ✓ Заинтересовано в оттачивании инструмента.
- ✓ Готово к обсуждениям.
- ✓ Работает на репутацию.
- ✓ Принимает решения относительно быстро.

Компания

- × Заинтересована в получении прибыли.
- × Фреймворк часто является побочным продуктом проектов компании.
- × Работает за деньги.
- × Часто откладывает принятие решений.

Почему BSD?

- Фреймворк – инструмент.
- Чтобы код «жил» его должны серьёзно использовать.
- Серьёзно = для получения прибыли.
- BSD не ограничивает коммерческое использование.

Сообщество

Yii

- Оттянул на себя часть сообщества Prado
- Не превратиться в helpdesk
- Повторяющиеся вопросы → wiki, поиск
- Мало пишут — не всегда плохо

В общем

- Интересные уникальные решения, рецепты
- Собрать лояльное ядро
- Обучить на личном примере, что для сообщества хорошо
- Оставить шанс показать себя

Команда Yii

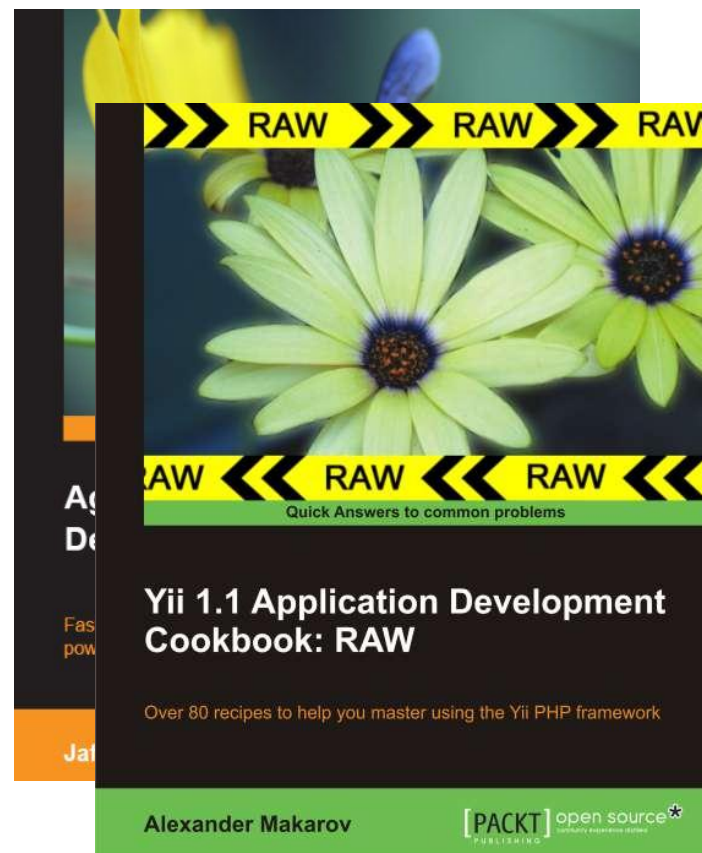
- 6 разработчиков.
- Технический писатель.
- ~50 переводчиков.
- Приоритеты:
 - Стабильность кода.
 - Надёжность кода.
 - Единый стиль.
- Количество разработчиков не растёт т.к. рост означает:
 - Больше времени на общение, меньше на код.
 - Возможные конфликты.

Use the force and read the source!

Нет уж, спасибо!

Документация

- Код без документации — мусор.
- Успешность проекта сильно зависит от документации.
- Пишем код — обновляем документацию.
- Всегда сразу документируем методы, классы и свойства.
- Важны примеры, но слишком много примеров — тоже плохо.



Почему код должен быть приятным?

- Чтобы проект доставлял разработчику не меньше удовольствия, чем пользователям.
- Чтобы не тратить неоправданно много времени на документацию.
- Деньги, время, мотивация.

Каким должен быть API?

- **Согласованный.** Всё в одном стиле.
- **Не сложный.** Сложность = много всего.
 - Магия, если она есть, должна обязательно быть однозначной и, по возможности, простой.
- **Гибкий.** Легко поменять поведение.
- **Документированный.**

- **Гибкость vs простота.**
 - Крайности. God object.
- **Решение:** фасад для скрытия деталей, адаптер для выравнивания API.

Готового идеала нет

- В больших проектах всегда приходится что-то подпиливать.
 - Как потом обновлять?
 - Как ничего не сломать?
- В фреймворк нельзя встроить всё.
 - Надо ещё и сторонний код пилить?
 - А вдруг не заработает?

Версии и обратная совместимость

1.1.x — обратно совместимые релизы.

- Небольшие документированные изменения.
- Новые возможности.
- Подходит для обновления уже начатых проектов.

1.x.0 — частичная несовместимость.

- Обновить проект возможно, но затратно.

x.0.0 — более свежие возможности без оглядки на обратную совместимость.

Поддержка старых версий

Проект

- Делается не один месяц, за это время выходит обратно несовместимая версия.
- Немедленное адаптирование под неё — потеря времени и денег.
- Возможно, стоит остановиться, но как быть с багами в фреймворке?

Фреймворк

- Поддерживать все старые версии нереально → ограничиваем время поддержки.
- Не перетаскиваем новые возможности в старые версии. Исправляем ошибки.

Инструментарий

- «Одноразовую» миграцию данных в большинстве случаев приходится делать много раз.
- Код надо тестировать.
- Однотипный код писать лень.
- IDE должны поддерживать код.
- Отладка — это важно.

Производительность и оптимизация

- Yii быстрый т.к. загружается только то, что нужно.
Используется SPL __autoload.
- Не подключаемся к БД, пока не понадобится, не парсим запрос пока не потребуют и т.д.
- Если код используется часто и много — производительность важнее фич.
- Если используется редко — производительность не так важна.



Yii — это хорошо!

Спасибо

- <http://yiiframework.com/>
- <http://yiiframework.ru/>
- <http://rmcreative.ru/>
- sam@rmcreative.ru
- Принимаются
 - Отзывы
 - Идеи
 - Предложения

