



Александр Макаров

Yii core team

Stay.com

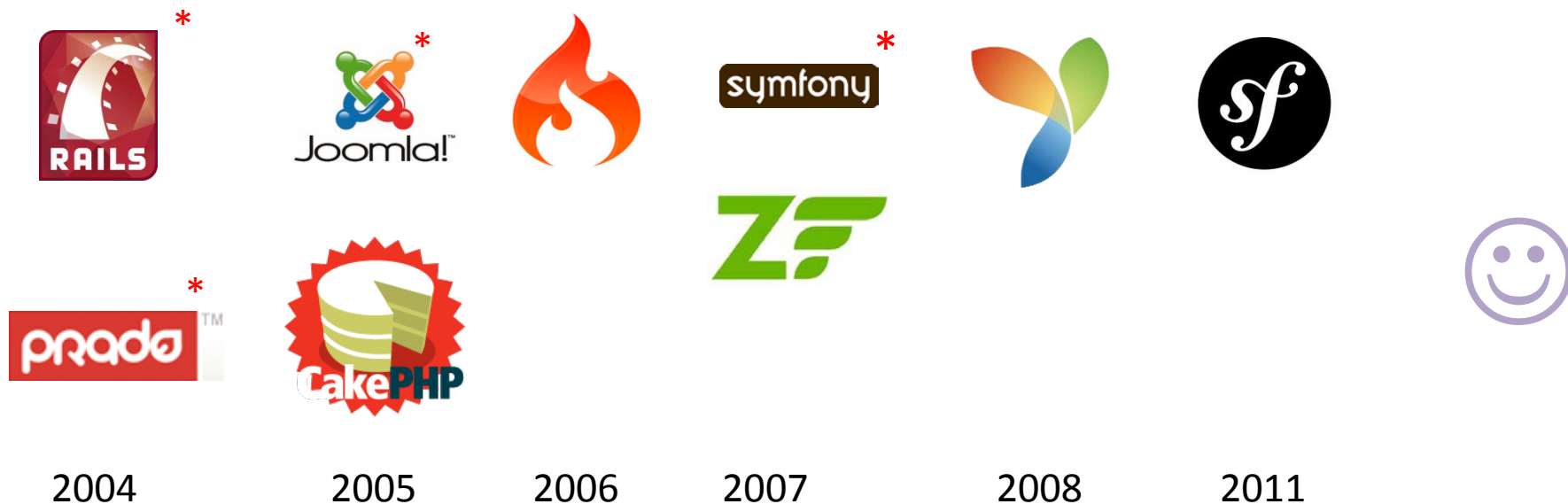
Yii?

- ✓ PHP5 MVC.
- ✓ Приятный API.
- ✓ DAO, AR, миграции.
- ✓ Построитель форм.
- ✓ Темы.
- ✓ Тесты.
- ✓ Документирован.
- ✓ Сообщество.
- ✓ BSD.
- ✓ Мощный кеш.
- ✓ RBAC, авторизация.
- ✓ Консоль.
- ✓ **Обработка ошибок, лог.**
- ✓ Gii.
- ✓ CLDR, I18n.
- ✓ Виджеты.



Что важно в фреймворке

- Легко изучать.
- Легко отлаживать и исправлять ошибки.
- **Не должен мешать** вашему коду или любому нормальному стороннему коду.
- Отличное сообщество.
- **Обратно совместимый и стабильный.**
- Не мешает расширять и настраивать.
- Приятен ;)



Prado, c 2004 → Yii 1.0, 2008 → Yii 1.1, 2010

Что? Где? Когда?

- Он **жив** 😊
- Он **развивается**.
- Он уже не так сильно далёк от публичной альфы... но всё ещё есть над чем поработать, поэтому пока никаких дат и прогнозов.

<https://github.com/yiisoft/yii2/wiki/Yii2-Development-Roadmap>

- Последний PHP **5.3.x+**
- **Namespace**
- **PSR-0**

- Лучше структура
- Меньше лишних классов
- Хорошее осталось

v2

ПРОИЗВОДИТЕЛЬНОСТЬ

Общая

- Лучше 1.1.
- "hello world" для веб **в два раза быстрее 1.1** без APC и на 10—20% быстрее с APC (новый PSR-0 загрузчик классов, namespaces, меньше классов).

AR

- Лучше 1.1.
- AR2 **кушает меньше памяти** и работает **быстрее**. В режиме asArray значительно.

СТРУКТУРА

Базовые классы

- Object
 - Геттеры/сеттеры (~1.1).
- Component
 - События.
 - Behavior (~1.1).

События

```
class MyComponent extends \yii\base\Component
{
    public function test($name)
    {
        $this->trigger('test', new \yii\base\Event($this, $name));
    }

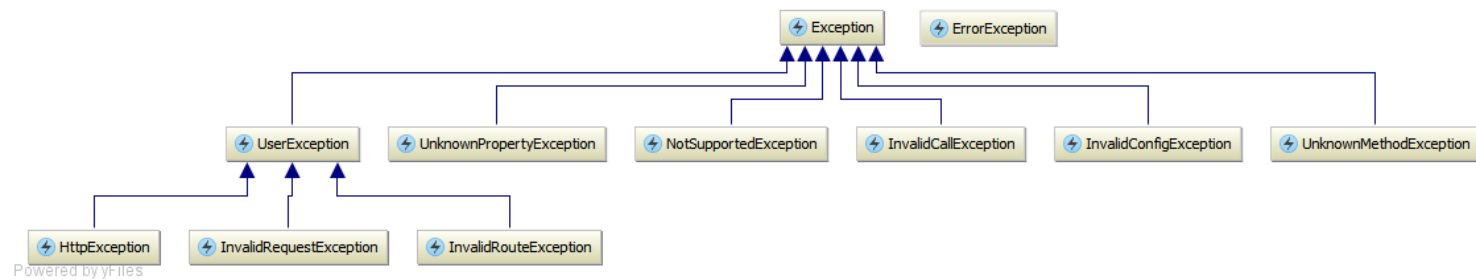
    public function hello($name)
    {
        echo 'Hello, '.$name.'!';
    }
}
```

```
$component = new MyComponent();
$component->on('test', function($event) {
    echo $event->sender->hello($event->data);
});
$component->test('Alex');
```

// out:

Hello, Alex!

Исключения



ОШИБКИ, ОТЛАДКА, ЛОГ

Ошибки

- Нефатальные ошибки конвертируются в исключения.
- Фатальные ошибки ловятся и показываются красиво.
- Фатальные ошибки пишутся в лог в подавляющем большинстве случаев.

Fatal Error (yii\base\Exception)

Call to undefined function calll_to_undefined()

W:\home\yii2.local\protected\controllers\SiteController.php(41)

```
29     $posts = $command->queryAll();
30
31     //$test = new Test();
32     //echo $test;
33
34     //     $x = str_repeat('x', 535822311);
35     //     $y = '';
36     //     while(-1) {
37     //         $y .= 'x';
38     //     }
39
40     //asdfw4;;
41     calll_to_undefined();
42     //throw new \yii\base\Exception('ollololo');
43     //trigger_error('test', E_USER_ERROR);
44
45     echo $this->render('index', array(
46         'posts' => $posts,
47     ));
48 }
49
50 public function actionTwig()
51 {
52     echo $this->render('renderer.twig', array('username' => 'Alex'));
53 }
```

Stack Trace

- #0 [D:\src\yii2\framework\base\Module.php(568): yii\base\Controller::runAction()]
- #1 [D:\src\yii2\framework\web\Application.php(36): yii\base\Module::runAction()]
- #2 [D:\src\yii2\framework\base\Application.php(213): yii\web\Application::processRequest()]
- #3 [W:\home\yii2.local\www\index.php(11): yii\base\Application::run()]

```
06 $config = require dirname(__DIR__).'protected/config/main.php';
07
08 require 'd:/src/yii2/framework/yii.php';
09 $basePath = dirname(__DIR__).'protected';
10 $app = new \yii\web\Application('webapp', $basePath, $config);
11 $app->run();
```

Перехват фатальных ошибок в PHP

- `register_shutdown_function() / error_get_last()`.
- `ini_set('display_errors', 0);`
- Стек можно получить из XDebug.
- Ошибки нехватки памяти. Резерв.

Особенности конвертирования ошибок в исключения

- `set_error_handler`, кидаем исключение.
- `set_exception_handler`, обрабатываем.
- Исключения в `__toString()`.

Отладка и лог

- Ошибки пишутся в лог всегда.
- В планах полноценная отладочная панелька.

РАСШИРЕНИЯ

Зависимости, сайт

- Composer.
 - Официальные расширения.
 - Рекомендуемые расширения.
 - [github/bitbucket](https://github.com/bitbucket).
-
- Несколько официальных расширений будут доступны сразу (Smarty, Twig).

МОДЕЛЬ

Модель

- Больше нет модели формы.
- `ArrayAccess`.
- Валидация.
- Массовое присваивание.

Модель

```
class LoginForm extends \yii\base\Model
{
    public $username;
    public $password;
}
```

Массовое присваивание

- У нас дыра в безопасности.
- Слава богу, хоть что-то у нас в безопасности...



AR

```
class User extends \yii\db\ActiveRecord
{
    public function table()
    {
        return 'tbl_user';
    }

    public function scenarios()
    {
        return array(
            'login' => array('username', 'password'),
            'register' => array('username', 'email',
'password'),
        );
    }
}
```

Валидация

```
function rules() {  
    return array(  
        // rule applied when corresponding field is "safe"  
        array('username', 'length', 'min' => 2),  
        array('first_name', 'length', 'min' => 2),  
        array('password', 'required'),  
  
        // rule applied when scenario is "signup" no matter if field is "safe" or not  
        array('hashcode', 'check', 'on' => 'signup'),  
    );  
}  
  
function scenarios() {  
    return array(  
        // on signup allow mass assignment of username  
        'signup' => array('username', 'password'),  
        'update' => array('username', 'first_name'),  
    );  
}
```

РАБОТА С БД

Command

```
Try {  
    $command = $db->createCommand($sql)-  
>queryAll();  
}  
catch(\yii\db\Exception) {  
    //  
}
```

Query

```
$query = new Query();  
$result = $query->select('*')->  
>from('post')->where('id = :id',  
array(':id' => 1))->createCommand()-  
>query();
```

AR

```
$customers = Customer::find()  
    ->where(array('status' => 1))  
    ->orderBy('age')  
    ->all();
```

```
$customer = Customer::find(10);
```

```
$customer = Customer::find(array('age' =>  
30, 'status' => 1));
```


AR relations

```
class Customer extends \yii\db\ActiveRecord
{
    public function getOrders()
    {
        return $this->hasMany('Order',
array('customer_id' => 'id'));
    }
}
```

```
$customer = Customer::find($id);
$orders = $customer->orders; // $orders is an
array of Order objects
```

AR link/unlink

```
$customer = Customer::find(1);  
$order = new Order;  
$order->subtotal = 100;  
$customer->link('orders', $order);
```

AR scope

```
class Customer extends \yii\db\ActiveRecord
{
    // ...

    /**
     * @param ActiveQuery $query
     */
    public function active($query)
    {
        $query->andWhere('status = 1');
    }
}

$customers = Customer::find()->active()->all();
```

AR scope

```
class Customer extends \yii\db\ActiveRecord
{
    // ...

    /**
     * @param ActiveQuery $query
     * @param integer $age
     */
    public function olderThan($query, $age = 30)
    {
        $query->andWhere('age > :age', array(':age' =>
$age));
    }
}

$customers = Customer::find()->olderThan(50)->all();
```

noSQL?

- Всякие варианты кеша.
- Не кэш:
 - Пробуем написать обёртку для Redis.
 - Планируем также пощупать Mongo.



WEB, КОНСОЛЬ, ШАБЛОНЫ

Консольное приложение

- Консольное и веб очень похожи.
- Консольная команда = контроллер.
- Консолька будет цветастая.
- Документация для help из phpdoc.

yīc app

- Как минимум два приложения
 - Простое
 - Посложнее
- Можно сделать свои шаблоны.

1 или 2?

Работайте на стабильном.

**1.1 поддерживается как минимум до
31 декабря, 2015.**

Вопросы?

- <http://yiiframework.com/>
- <https://github.com/yiisoft/yii2/wiki/Yii2-Development-Roadmap>
- sam@rmcreative.ru
- @sam_dark

