Yii^2 What's new?
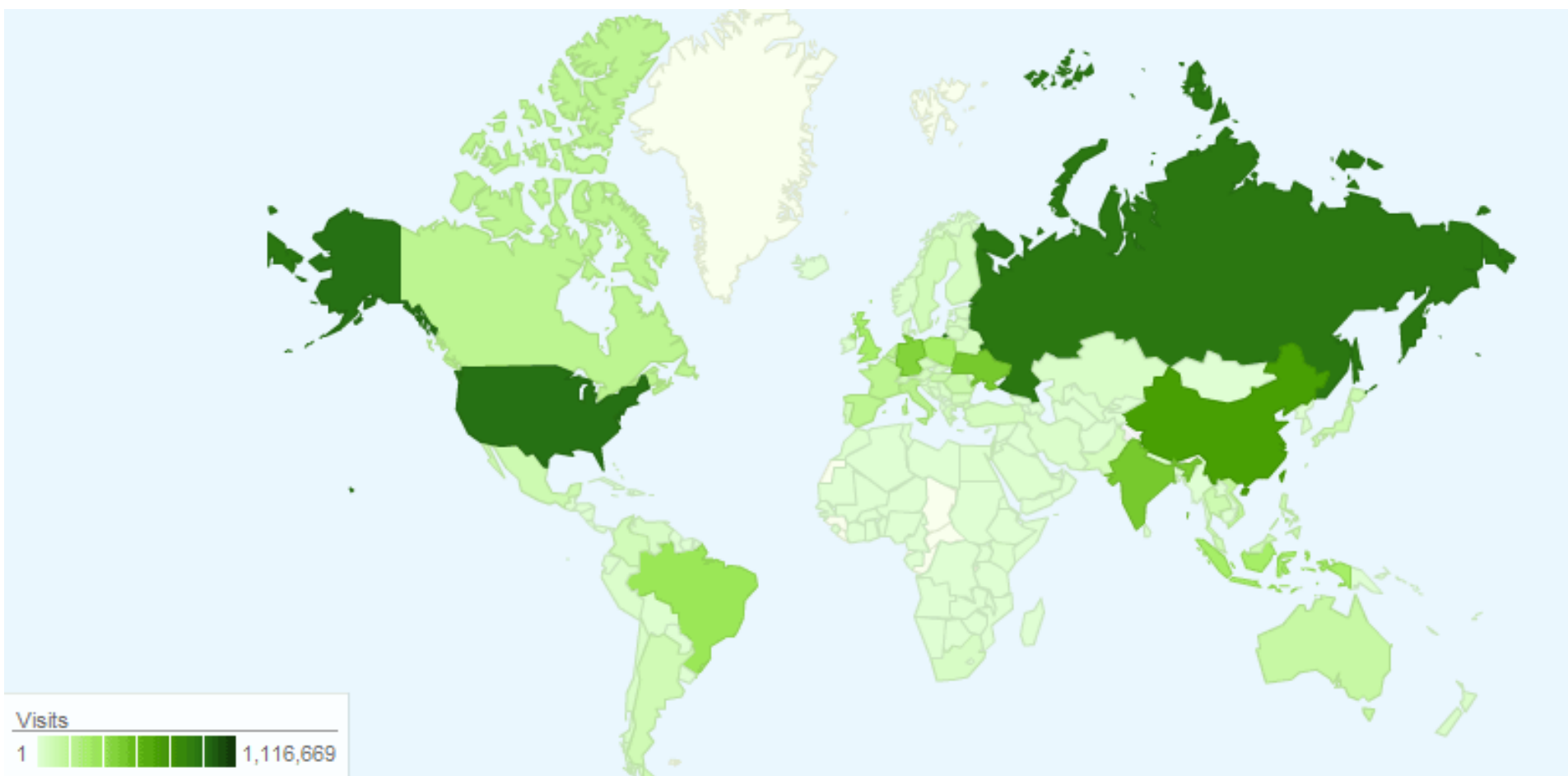
Alexander Makarov,

Yii core team

yii conf

Some statistics

Visits

1   1,116,669

20,000   20,000

10,000   10,000

Oct 1   Oct 1   Apr 1   Oct 1   Apr 1   Oct 1

▼

75.06%

24.94%
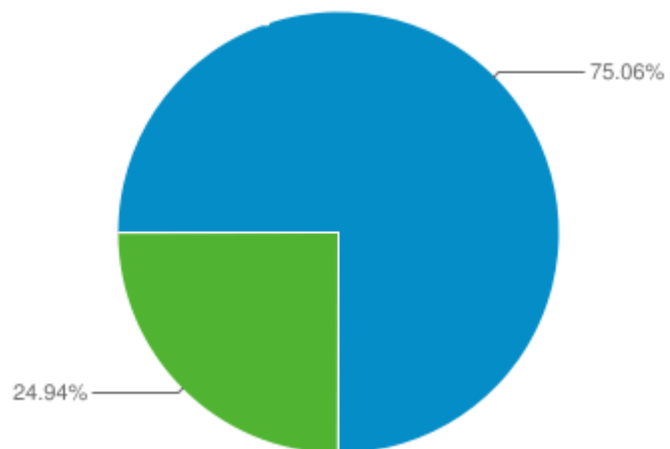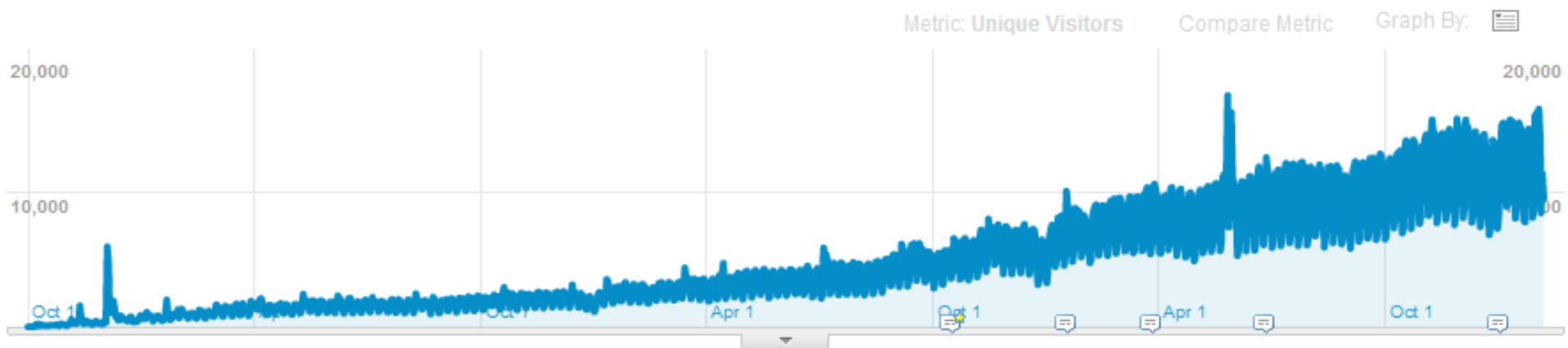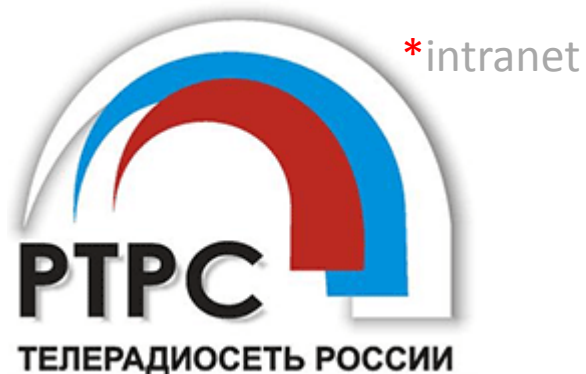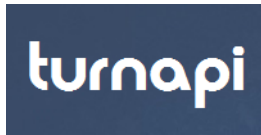
Who uses Yii?

Why?

1. Well-balanced
2. Stable
3. Flexible
4. Well-documented

2011 — beginning of 2012 events

What happened?

- 5 stable Yii 1.1 releases
- Yii 1.1 Application Development Cookbook
- Yii for Eclipse PDT, CodeLobster
- Yii → GitHub

- Events:
  - Yii beer party
  - YiiTalk
  - YiiConf

First two days

—348 watches

—61 forks
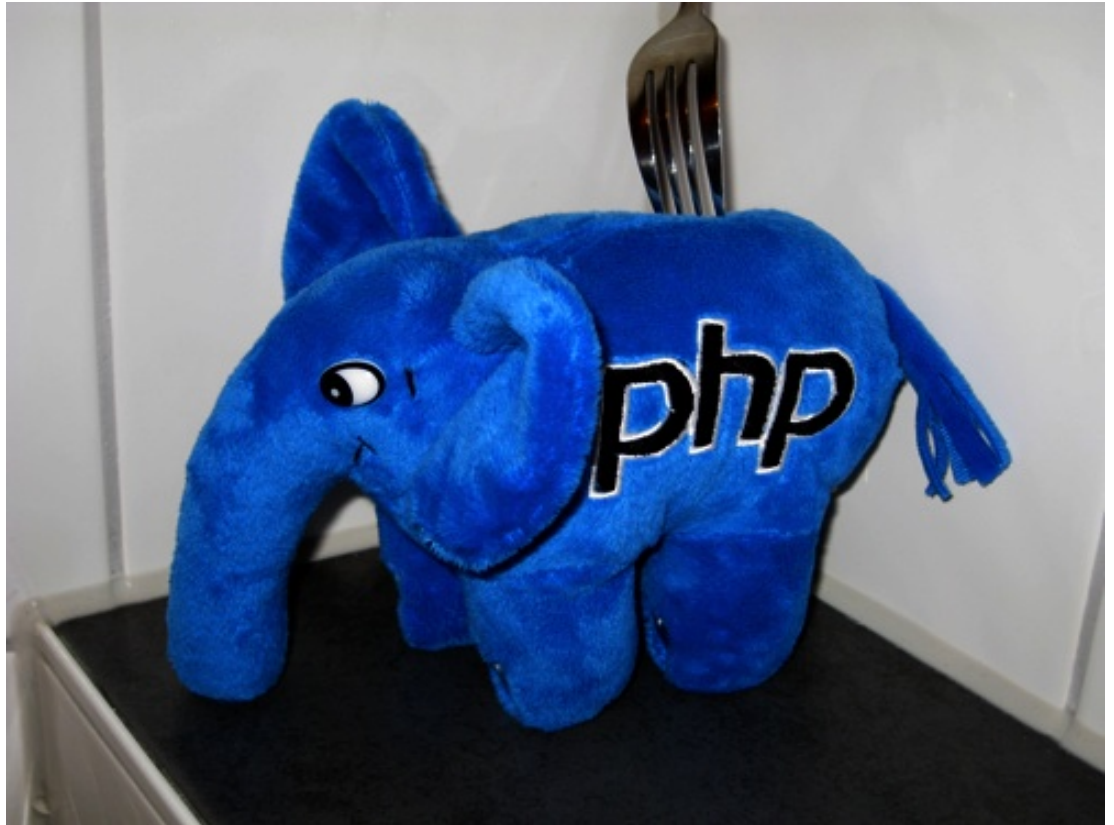
Now

—1134 watches

—240 forks
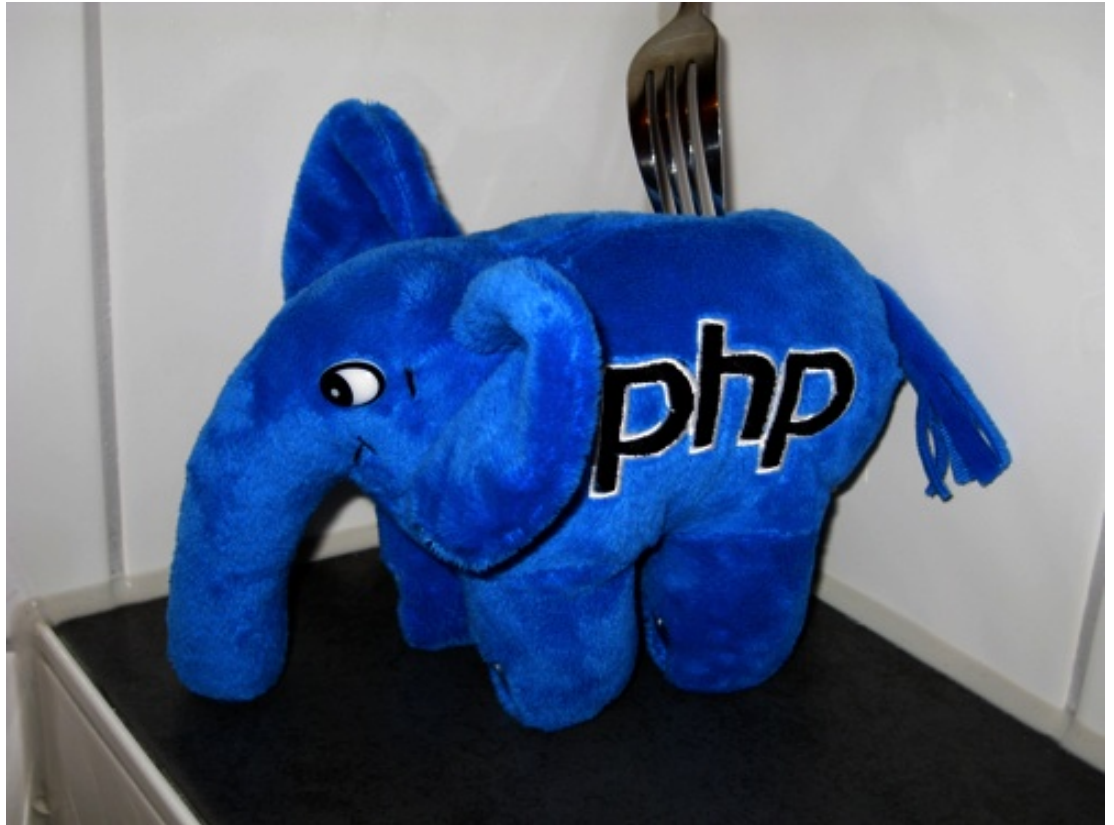
Yii 1.1.11 will be quite interesting release ;)

What's bad in Yii 1.1?

- AR (finder and record are the same object, some API).

- Some classes are in strange places.

- More small things.

The biggest issue is BC

2004   2005   2006   2007   2008   2011
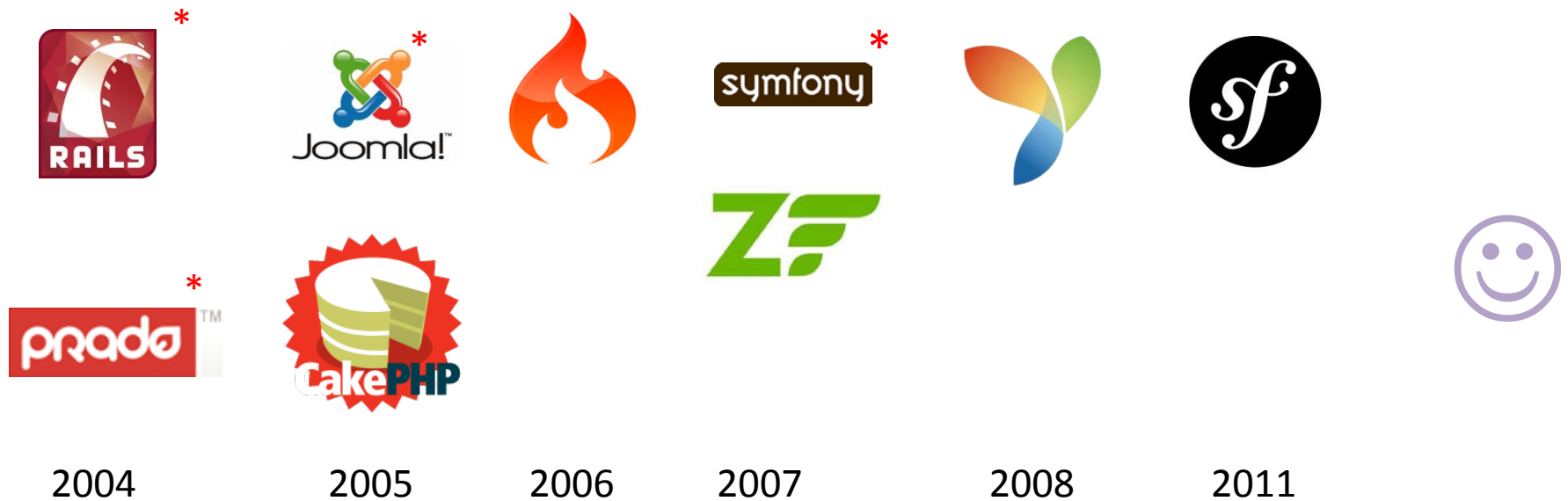
Prado, from 2004 → Yii 1.0, 2008 → Yii 1.1, 2010

Yii 2 team

—3 active core developers: qiang, samdark, mdomba.

—github.

—PHP 5.3.8+

—All classes are namespaced (\yii) and w/o prefix

—PSR-0

—Better structure

—Less entities

—Keep good ideas

v2

Documentation

—**Larry Ullman**, author of 22 excellent IT-books and a great article series about Yii will write a book about Yii2 + will participate in official documentation writing.

—API docs will be at least same quality as 1.1.

—**Code style guide**.

—There's a plan to release documentation tool to the public.

Yii2: base

— Aliases in form of @yii/base/Component

— CComponent → Object + Component

— SPL replaced most of collections

— Removed CFormModel. Now you can use Model directly.

```
class MyComponent extends
\yii\base\Object
{
  public $x;
  public function __construct($a, $b)
  {
      //…
  }
}

$component = MyComponent::newInstance(
  array('x'=>10),
  'a', 'b'
);
```

Yii2: View Object

- render(), widget(), beginCache() → viewObject
- In a View: $owner = class that uses view
- $this = View.
- No need for renderer.
- Can be used in console applications.
- CHtml is still there.

Yii2: events

```
$post->on('add',
function($event) { ...
});

$post->trigger('add',
new Event($this));

$post->off('add',
$callback);

$handlers = $post-
>getEventHandlers('add'
);
```

- No need to explicitly declare before raising
- jQuery-like syntax
- Behaviors instead of filters

Yii2: Query object

```
// Query object
$query = new Query;
$query->select('id')->from('tbl_customer')->limit(10);
$command = $db->createCommand($query);
$this->assertEquals("SELECT `id` FROM `tbl_customer`
LIMIT 10", $command->sql);

// array
$command = $db->createCommand(array(
    'select' => 'name',
    'from' => 'tbl_customer',
));
```

Yii2: AR

```
$customer = Customer::find(2)
        ->active()
        ->one();
$customer->name = 'Qiang';
$customer->save();

$customers = Customer::find()
        ->order('id')
        ->asArray(true)
        ->all();
```

– Finder / Model
– Can create your own finder
– ~~::model()~~
– Auto quoting.
– Method chains.

Yii2: AR

```php
$postFinder = Post::find()
  ->where(array(
    'active' => true
));
if($isPrivate) {
  $postFinder->addWhere(array(
    'createdBy' => $userId,
  ));
}
$posts = $postFinder
  ->mergeWith($anotherFinder)
  ->all();
```

- ~~Criteria~~
- Can merge two finders
- Can add conditions on the fly

Yii2: AR

- tableName(), relations(), scopes() = static.
- Relations are HAS_ONE, HAS_MANY.
- link = FKs
- via = through
- Anonymous functions for scopes.
- "@." and "?" tokens: own table, external table.

```
class Customer extends ActiveRecord {
    const STATUS_ACTIVE = 1;

    public static function tableName() {
        return 'tbl_customer';
    }

    public static function relations() {
        return array(
            'orders:Order[]' => array(
                'link' => array('customer_id'
=> 'id'),
            ),
        );
    }

    public static function scopes() {
        return array(
            'active' => function($q) {
                return $q-
>andWhere('@.`status` = ' . self::STATUS_ACTIVE);
            },
        );
    }
}
```

Yii2: AR

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

foreach (Customer::find()
as $customer)

$count = Customer::count()
->value();
```

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

$customers =
Customer::find()
->where('name like :name',
array(
  ':name' => '%customer%'
))->order('id')->all();
```

TODO (if there will be enough time)

- HTTP (CURL) wrapper
- Package manager
- Mailer
- Twitter Bootstrap
- Debug toolbar
- Console requirements
- More helpers

- jQueryUI-based widgets
- Commercial support
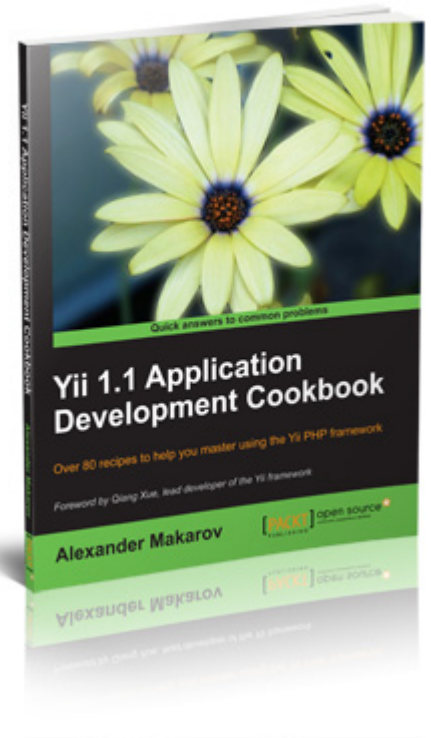
1 or 2?

Don't wait. Work with stable one.

When?

Before alpha code will be put into public github repository we need to finish at least these things→

- Base for caching
- Base for i18n
- Controller + webapp
- Base for widgets
- URL manager

Want more info?

- http://www.yiiframework.com/forum/index.php/forum/42-design-discussions-for-yii-20/

Questions?

yiiframework.com

yiiframework.ru

rmcreative.ru