

1.2 @ OCPHP

August 9, 2007

Why CakePHP?

- MVC architecture
- Convention over Configuration
- Flexible and Extensible
- Write less, do more
- Open Source, Large Community
- MIT License

CakePHP the right way



Think twice, code once
Loose couple

Hot out of the oven

- Behaviors
- “With” Associations
- Enhanced Validation
- Forms
- Pagination
- i18n + l10n
- Acl
- Router
- Email
- Security
- Authentication
- Configure
- Cache Engines
- Console

Behaviors

- extend models
- encapsulate reusable functionality
- use callbacks for automagic

```
<?php
class CoolBehavior extends ModelBehavior {

    var $settings = array();

    function setup(&$model, $config = array()) {
        $this->settings = $config;
    }
    /* Callbacks */
    function beforeFind(&$model, $query) { }

    function afterFind(&$model, $results, $primary) { }

    function beforeSave(&$model) { }

    function afterSave(&$model, $created) { }

    function beforeDelete(&$model) { }

    function afterDelete(&$model) { }

    function onError(&$model, $error) { }

    /* Custom Methods */
    function myCustomMethod(&$model, $data = array()) {
        return array_merge($model->data, $data);
    }
}
?>
```

```
<?php
class Post extends AppModel {

    var $actsAs = array('Cool'=>array('field'=>'cool_id'));

    function doSomething($cool = null) {
        $this->myCustomMethod(array('cool_id'=> $cool));
    }

}
?>
```

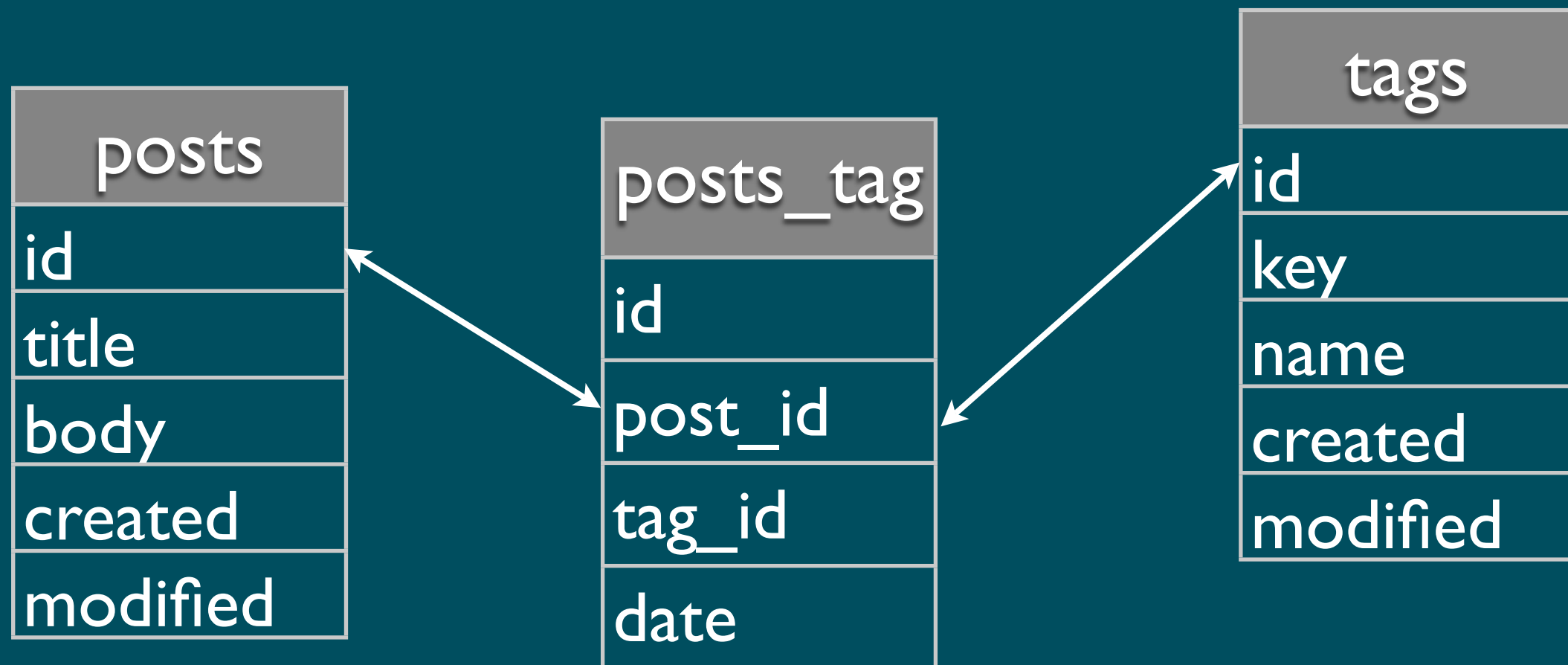
```
<?php
class PostsController extends AppController {
    var $name = 'Posts';

    function index() {
        $this->set('posts', $this->Post->doSomething(5));
    }

}
?>
```

“With” Associations

- enables extra fields in join tables
- access to model for join table



Adding “With”

```
<?php
class Post extends AppModel {

    var $hasAndBelongsToMany = array(
        'Tag' => array(
            'className' => 'Tag',
            'with' => 'TaggedPost',
        )
    );

    function beforeSave() {
        if(!empty($this->data['Tag'])) {
            $this->TaggedPost->save($this->data['Tag']);
        }
    }

}
?>
```

Using “With”

```
<?php
class PostsController extends AppController {
    var $name = 'Posts';

    function tags() {
        $this->set('tags', $this->Post->TaggedPost->findAll());
    }
}
?>
```

```
<?php
foreach ($tags as $tag) :
    echo $tag['Post']['title'];

    echo $tag['Tag']['name'];

    echo $tag['TaggedPost']['date'];

endforeach;
?>
```

Validation is King

- | | | |
|--------------------------------|-----------------------------|-------------------------------|
| ➤ alphaNumeric | ➤ email | ➤ number |
| ➤ between | ➤ equalTo | ➤ numeric |
| ➤ blank | ➤ file | ➤ phone |
| ➤ cc | ➤ ip | ➤ postal |
| ➤ comparison | ➤ maxLength | ➤ ssn |
| ➤ custom | ➤ minLength | ➤ url |
| ➤ date | ➤ money | ➤ userDefined |
| ➤ decimal | ➤ multiple | |

Using Validation

```
<?php
class Post extends AppModel {

    var $validate = array(
        'title' => array(
            'required' => VALID_NOT_EMPTY,
            'length' => array( 'rule' => array('maxLength', 100))
        ),
        'body' => VALID_NOT_EMPTY
    );

}
?>
```

Forms made simple

```
<?php

echo $form->create('Post');

echo $form->input('title', array('error' => array(
    'required' => 'Please specify a valid title',
    'length' => 'The title must have no more than 100 characters'
)
));

echo $form->input('body', array('error' => 'Please specify a valid body'));

echo $form->end('Add');
?>
```

More about Forms

```
<?php
    /* Change the action */
    echo $form->create('User', array('action'=> 'login'));
?>
```

```
<?php
    /* Change the type for uploading files */
    echo $form->create('Image', array('type'=> 'file'));
    echo $form->input('file', array('type'=> 'file'));
    echo $form->end('Upload');
?>
```

```
<?php
    /* the three line form */
    echo $form->create('Post');
    echo $form->inputs('Add a Post');
    echo $form->end('Add');
?>
```

Pagination

```
<?php
class PostsController extends AppController {
    var $name = 'Posts';

    var $paginate = array('order'=> 'Post.created DESC');

    function index() {
        $this->set('posts', $this->paginate());
    }
}
?>
```

```
<?php
echo $paginator->counter(array(
    'format' => 'Page %page% of %pages%, showing %current% records out of
    %count% total, starting on record %start%, ending on %end%'
));
echo $paginator->prev('<< previous', array(), null, array('class'=>'disabled'));
echo $paginator->numbers();
echo $paginator->sort('Created');
echo $paginator->next('next >>', array(), null, array('class'=>'disabled'));
```

i18n/L10n

- built in gettext for static strings
- TranslationBehavior for dynamic data
- console extractor

gettext style

```
<?php
/* echo a string */
__(‘Translate me’);

/* return a string */
$html->link(__(‘Translate me’, true), array(‘controller’=>‘users’));

/* by category : LC_CTYPE, LC_NUMERIC, LC_TIME, LC_COLLATE, LC_MONETARY, LC_MESSAGES and LC_ALL.*/
__c(‘Translate me’, ‘LC_CTYPE’);

/* get plurals by number*/
__n(‘Translate’, ‘Translates’, 2);

/* override the domain.*/
__d(‘posts’, ‘Translate me’);

/* override the domain by category.*/
__dc(‘posts’, ‘Translate me’, ‘LC_CTYPE’);
```

Translation Behavior

```
<?php
class Post extends AppModel {

    var $actsAs = array('Translation'=> array('title','body'));

}
?>
```

```
CREATE TABLE i18n (
    id int(10) NOT NULL auto_increment,
    locale varchar(6) NOT NULL,
    i18n_content_id int(10) NOT NULL,
    model varchar(255) NOT NULL,
    row_id int(10) NOT NULL,
    field varchar(255) NOT NULL,
    PRIMARY KEY (id),
    KEY locale (locale),
    KEY i18n_content_id (i18n_content_id),
    KEY row_id (row_id),
    KEY model (model),
    KEY field (field)
);
```

```
CREATE TABLE i18n_content (
    id int(10) NOT NULL auto_increment,
    content text,
    PRIMARY KEY (id)
);
```

Acl

- abstracted for maximum compatibility
- component for controller
- behavior for model

Acl behavior

```
<?php
class User extends AppModel {

    var $actsAs = array('Acl'=> 'requester');

    function parentNode() {}

    function bindNode($object) {
        if (!empty($object['User']['role'])) {
            return 'Roles/' . $object['User']['role'];
        }
    }

    function roles() {
        return $this->Aro->findAll();
    }
}
?>
```

```
<?php
class Post extends AppModel {

    var $actsAs = array('Acl'=> 'controlled');

    function parentNode() {}

    function bindNode($object) {}
}
?>
```

Routing

- reverse routes
- magic variables
- named parameters
- parseExtensions
- REST resource mapping

Reversing Routes

```
<?php
Route::connect('/', array('controller'=>'pages', 'action'=>'display', 'home'));
?>
```

```
<?php
echo $html->link('Go Home', array('controller'=> 'pages', 'action'=> 'display', home));
?>
```

Magic Variables

`$Action` => Matches most common action names

`$Year` => Matches any year from 1000 to 2999

`$Month` => Matches any valid month with a leading zero (01 - 12)

`$Day` => Matches any valid day with a leading zero (01 - 31)

`$ID` => Matches any valid whole number (0 - 99999.....)

```
Router::connect(
   ('/:controller/:year/:month/:day',
    array('action' => 'index', 'day' => null),
    array('year' => $Year, 'month' => $Month, 'day' => $Day)
);
```

```
Router::connect(
   ('/:controller/:id',
    array('action' => 'index', 'id' => '1'), array('id' => $ID)
);
```

Named Args

<http://cakephp.org/posts/index/page:2/sort:title>

```
<?php
class PostsController extends AppController {
    var $name = 'Posts';

    function index() {
        $this->passedArgs['page'];
        $this->passedArgs['sort'];
    }
}
?>
```

Parsing Extensions

```
/* get everything */  
Router::parseExtensions();  
/* more fine grained */  
Router::parseExtensions('xml', 'html');
```

maps views to unique locations

<http://cakephp.org/posts/index.rss>

```
Router::parseExtensions('rss');
```

```
<?php //app/views/posts/rss/index.ctp  
echo $rss->items($posts, 'transformRSS');  
function transformRSS($post) {  
    return array(  
        'title'           => $post['Post']['title'],  
        'link'            => array('url'=>'/posts/view/'. $post['Post']['id']),  
        'description'     => substr($post['Post']['body'], 0, 200),  
        'pubDate'         => date('r', strtotime($post['Post']['created'])),  
    );  
}  
?>
```

Email

```
$this->Email->to = $user['User']['email'];
$this->Email->from = Configure::read('Site.email');
$this->Email->subject = Configure::read('Site.name').' Password Reset URL';

$content[] = 'A request to reset your password has been submitted to '.Configure::read('Site.name');
$content[] = 'Please visit the following url to have your temporary password sent';
$content[] = Router::url('/users/verify/reset/'.$token, true);

if($this->Email->send($content)) {
    $this->Session->setFlash('You should receive an email with further instruction shortly');
    $this->redirect('/', null, true);
}

/* Sending HTML and Text */
$this->Email->sendAs = 'both';
$this->Email->template = 'reset_password';

/* HTML templates */
//app/views/elements/email/html/reset_password.ctp

/* TEXT templates */
//app/views/elements/email/text
```

Security

HTTP Auth

```
<?php
class PostsController extends AppController {
    var $name = 'Posts';

    var $components = array('Security');

    function beforeFilter() {
        $this->Security->requireLogin('add', 'edit', 'delete');
        $this->Security->loginUsers = array('cake'=>'tooeasy');
    }
}
?>
```

Authentication

```
<?php
class PostsController extends AppController {
    var $name = 'Posts';

    var $components = array('Auth');
}
?>
```

- authenticate any model: User, Member, Editor, Contact
- authorize against models, controllers, or abstract objects
- automate login

Auth setup

```
<?php
class PostsController extends AppController {
    var $name = 'Posts';

    var $components = array('Auth');

    function beforeFilter() {
        $this->Auth->userModel = 'Member';
        $this->Auth->loginAction = '/members/login';
        $this->Auth->loginRedirect = '/members/index';
        $this->Auth->fields = array('username' => 'username', 'password' => 'pword');

        $this->Auth->authorize = array('model'=>'Member');
        $this->Auth->mapActions(array(
            'read' => array('display'),
            'update' => array('admin_show', 'admin_hide'),
        ));
    }
}
```

Authorize a Model

```
<?php
class Member extends Model {
    var $name = 'Member';

    function isAuthorized($user, $controller, $action) {

        switch ($action) {
            case 'default':
                return false;
            break;
            case 'delete':
                if($user[['Member']]['role'] == 'admin') {
                    return true;
                }
            break;
        }
    }
}
?>
```

Authorize a Controller

```
<?php
class PostsController extends AppController {
    var $name = 'Posts';

    var $components = array('Auth');

    function beforeFilter() {
        $this->Auth->authorize = 'controller';
    }

    function isAuthorized() {
        if($this->Auth->user('role') == 'admin') {
            return true;
        }
    }
}
?>
```

Authorize CRUD

```
<?php
class PostsController extends AppController {
    var $name = 'Posts';

    var $components = array('Auth');

    function beforeFilter() {
        $this->Auth->authorize = 'crud';
        $this->Auth->actionPath = 'Root/';
    }
}
?>

$this->Acl->allow('Roles/Admin', 'Root');
$this->Acl->allow('Roles/Admin', 'Root/Posts');
```

Configure

```
<?php
```

```
Configure::write('Oc.location', 'Buddy Group');
```

```
Configure::write('Oc', array('location' => 'Buddy Group'));
```

```
Configure::read('Oc.location');
```

```
Configure::store('Groups', 'groups',  
                array('Oc' => array('location' => 'Buddy Group')));
```

```
Configure::load('groups');
```

```
?>
```

Cache Engines

- File
- APC
- Memcache
- xcache
- Model

```
Cache::engine('File', [settings]);

$data = Cache::read('post');
if(empty($data)) {
    $data = $this->Post->findAll();
    Cache::write('post', $data);
}
```

Console

- Makes writing Command Line Interfaces (CLI) cake
- Shells
- Tasks
- Access to models

Core Shells

- Acl
- Extractor
- Bake
- Console
- API

The Bake Shell

- Tasks
 - Model – `cake bake model Post`
 - Controller – `cake bake controller Posts`
 - View – `cake bake view Posts`
 - DbConfig – `cake bake db_config`
 - Project – `cake bake project`

Custom view templates, oh yeah!

Your Own Shell

```
<?php
class DemoShell extends Shell {

    var $uses = array('Post');

    var $tasks = array('Whatever', 'Another');

    function initialize() {}

    function startup() {}

    function main() {
        $this->out('Demo Script');
    }

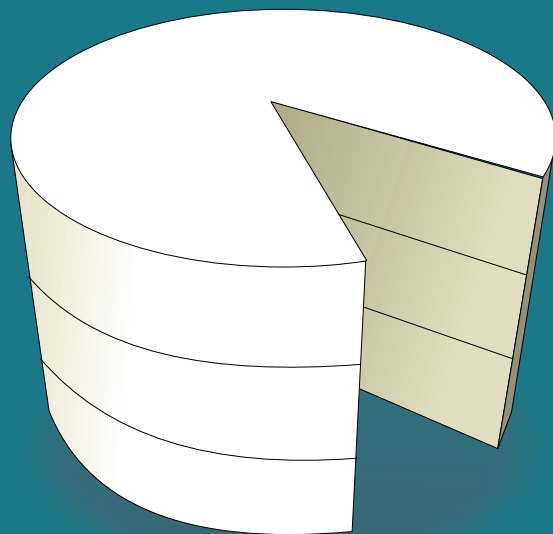
    function something() {
        $this->Whatever->execute();
    }

    function somethingElse() {
        $posts = $this->Post->findAll();
        foreach($posts as $post) {
            $this->out('title : ' . $post['Post']['title']);
        }
    }
}
?>
```

More....

- > Xml reading and writing
- > HttpSockets
- > Themes
- > Cookies
- >

Questions?



CakePHP