The CakePHP Media Plugin

Marius Wilms aka David Persson

July 12th 2009

Myself

- Fine Art Student at Braunschweig University of Art
- 8 years PHP
- 2.5 years CakePHP
- CakePHP core dev since march

This Talk

- Handling file uploads
- Generating different versions of files
- Creating the markup to embed it
- Some advanced usage

The Plugin

A plugin for CakePHP enabling transfer/manipulation/embedding of files in 23 ways.

- Why?
- Stable: 0.50 / Unstable: 0.60alpha
- CakePHP 1.2.x.x and PHP >= 5.2.0 required
- Source: https://github.com/davidpersson/media/tree
- Docs: http://wiki.github.com/davidpersson/media

Setting It Up

Get a fresh copy

```
git clone git://github.com/davidpersson/media.git app/plugins/media
```

Loading the configuration

app/config/bootstrap.php

```
require APP . 'plugins/media/config/core.php';
```

Initializing the directories

```
cake/console/cake media init
```

```
chmod -R a+rwX app/webroot/media/{transfer,filter}
```

Introduction

- Thousands of components, behaviors, libaries
- If not done right: Potential security hole
- Trickier than one would think

HTML Form & Uploads

Modifying the template

app/views/movies/add.ctp

```
echo $form->create('Movie', array('type' => 'file'));
// ...
echo $form->input('file', array('type' => 'file'));
// ...
```

A Request

POST /movies/add

Header:
Content−Type: multipart/form−data; boundary=12591
Content−Length: 32888

Body:
−−12591
Content−Disposition: form−data; name="‘_method"’

POST
−−12591
Content−Disposition: form−data; name="‘data[Movie][title]"’

Demo
−−12591
Content−Disposition: form−data; name="‘data[Movie][file]"’; filename="‘clipboard.jpg"’
Content−Type: image/jpeg

... data ...
−−12591−−

200 OK

Resulting data

MoviesController::$data

```
Array
(
    [Movie] => Array
    (
        [ title ] => Demo
        [ file ] => Array
            (
                [name] => clipboard.jpg
                [type] => image/jpeg
                [tmp_name] => /private/var/tmp/php4zN7BY
                [ error ] => 0
                [ size ] => 32405
            )
    )
)
```

Validation

One shouldn't trust

- Original filename (weird characters)
- Original file extension
- Submitted MIME type

Uploading An Executable: The Request

POST /movies/add

Header:
Content−Type: multipart/form−data; boundary=cAkePhPrUlEz
Content−Length: 32888

Body:
−−cAkePhPrUlEz
Content−Disposition: form−data; name="‘_method"’

POST
−−cAkePhPrUlEz
Content−Disposition: form−data; name="‘data[Movie][title]"’

Fake
−−cAkePhPrUlEz
Content−Disposition: form−data; name="‘data[Movie][file]"’; filename="‘ex.php"’
Content−Type: image/jpeg

<?php **echo** "‘Hello PHP."’ ?>
−−cAkePhPrUlEz−−

200 OK

Uploading An Executable: Resulting Data

MoviesController::$data

```
Array
(
    [Movie] => Array
    (
        [ title ] => Fake
        [ file ] => Array
            (
                [name] => ex.php
                [type] => image/jpeg
                [tmp_name] => /private/var/tmp/phpn4BPKr
                [ error ] => 0
                [ size ] => 26
            )
    )
)
```

Lessons Learned

- Do not trust the submitted data: Validate!
- Store uploaded files outside the webroot
- Serve only i.e. resized images
- Make the upload location not guessable i.e. use an UUID

The Transfer Behavior

What can it do for me?

- Handles several kinds of uploads
  - HTTP POST
  - Remote files via HTTP
  - Local files
- Validates transfers (8 new validation rules)

Usage

Modifying the model

app/models/movie.php

```
class  Movie extends AppModel {
    var  $actsAs = array('Media.Transfer');
}
```

Relocating The Transfer Directory

Updating the config

app/config/bootstrap.php

```
define('MEDIA_TRANSFER', APP . 'transfer' . DS);
define('MEDIA_TRANSFER_URL', false);

require APP . 'plugins/media/config/core.php';
```

Reinitializing the directories

```
cake/console/cake media init

chmod -R a+rwX app/transfer
```

Validation

Methods

- location
- size
- pixels
- extension
- mimeType
- 3 (not so interesting methods more) ...

Validation

Adding rules

app/models/movie.php

```
class Movie extends AppModel {
    var $actsAs = array('Media.Transfer');

    var $validate = array(
        'file' => array(
            'mimeType' => array(
                'rule' => array('checkMimeType', false, array('image/jpeg', 'image/png'))
            ),
            'size' => array(
                'rule' => array('checkSize', '5M')
            )
    ));
}
```

Reloca Transfer Demo

demo.sh reloca_transfer exploit

`http://localhost/workspace/media_demo/reloca_transfer/media/transfer/gen/ex.php`

`http://localhost/workspace/media_demo/reloca_transfer/movies/add`

Introduction

- A common task
- Need multiple libraries, no generic interface
- Helps securing transferred files (i.e. stripping comment tags)

The Media Behavior

What can it do for me?

- Generates versions of files
- Retrieves file metadata transparently (not covered)
- Couples files with database records (later)

Usage

Adding the behavior (chaining)

app/models/movie.php

```
class  Movie extends AppModel {
    var  $actsAs = array(
        'Media.Transfer',
        'Media.Media'
    );

    // ...
```

Usage

Overwritting default filter config

app/config/bootstrap.php

```
define('MEDIA_TRANSFER', APP . 'transfer' . DS);
define('MEDIA_TRANSFER_URL', false);

require APP . 'plugins' . DS . 'media' . DS . 'config' . DS . 'core.php';

Configure::write('Media.filter.image', array(
    's'    => array('convert' => 'image/png', 'fitCrop' => array(100, 100)),
    'm'    => array('convert' => 'image/png', 'fit' => array(300, 300)),
));
Configure::write('Media.filter.video', array(
    's'    => array('convert' => 'image/png', 'fitCrop' => array(100, 100)),
    'm'    => array('convert' => 'video/ogg', 'fit' => array(300, 300)),
));
```

Medium Class

An instance represents a single medium file.

- Manipulation
- Information

Available Adapters

- Gd
- Imagick (native & shell)
- Ffmpeg (native)
- GetId3
- PearMp3 & -Text

Contribute more adapters!

Manipulation

... is expensive.

When?

- Synchron (on upload)
- Asynchron
- Mixed

Possible criteria

- Version
- Medium type
- File size

Yet another callback: beforeMake

- Control which version gets generated when
- Delegate generation using i.e. a message queue
- Utilize custom manipulation methods

Yet another callback: beforeMake

Updating the model

app/models/movie.php

```
class Movie extends AppModel {
    // ...

    function beforeMake($file, $process) {
        extract($process);

        if ($version == 'm') {
            return ClassRegistry::init('Queue.Job')->put(compact('file', 'process'));
        }
    }
}
```

Return

- true: handled, version has been generated
- false or null: unhandled, behavior generates version

Introduction

- Multiple medium types
- Each one needs different markup
- Long paths

The Medium Helper

What can it do for me?

- Unified interface for easy embedding and linking
- Partial and dynamic paths
- Honor protected transfer directory

Partial Paths Support

- Works for all helper methods
- Dynamically expanded

css/cake.generic	/full/path/to/static/css/cake.generic.css
s/img/logo	/full/path/to/filter/s/static/img/logo.png
transfer/img/cern	/full/path/to/transfer/img/cern.jpg
m/transfer/img/cern	/full/path/to/filter/m/transfer/img/cern.png

Creating The Correct Markup

Adding the helper

app/controllers/movies_controller.php

```
class MoviesController extends AppController {
    var $helpers = array('Media.Medium');

    // ...
}
```

Creating The Correct Markup

Modifying the template

app/views/movies/view.ctp

```
<h2>Movie</h2>
    <dl>
        <dt>Id</dt>
        <dd><?php echo $movie['Movie']['id']; ?></dd>
        <dt>Title</dt>
        <dd><?php echo $movie['Movie']['title'];  ?></dd>
        <dt>File</dt>
        <dd><?php echo $medium->file($movie['Movie']); ?></dd>
        <dt>Preview</dt>
        <dd><?php echo $medium->embed($medium->file('filter/s', $movie['Movie'])); ?></dd>
    </dl>
```

Introduction

- Everything comes together

The Attachment Model

What does it provide?

- Central place for attached files
- Coupling of files with database records
- Already prepared with validation rules
- Other (multiple) models can relate to it (polymorphic)

Usage

Supplied model

app/plugins/media/models/attachment.php

```
class Attachment extends AppModel {
    var $actsAs = array(
            // ...
            'Media.Transfer' => array(
                'trustClient'       => false,
                'baseDirectory'     => MEDIA_TRANSFER,
                'destinationFile'   => ':Medium.short::DS::Source.basename:',
                'createDirectory'   => true,
                ),
            'Media.Media' => array(
                'metadataLevel'     => 2,
                'makeVersions'      => true,
                'filterDirectory'   => MEDIA_FILTER,
                ),
            );
    // ...
}
```

```
cake schema run create -path app/plugins/media/config/sql/ -name Media
```

Usage

Relationship

app/models/movie.php

```
class  Movie extends AppModel {
    var  $hasMany = array(
        'Attachment' => array(
            'className' => 'Media.Attachment',
            'foreignKey' => 'foreign_key',
            'conditions' => array('model' => 'Movie'),
            'dependent' => true,
     ));
}
```

Usage

Modfying the controller action

app/controllers/movies_controller.php

```
class MoviesController extends AppController {
    // ...
    function add() {
        if (!empty($this->data)) {
            $this->Movie->create();
            if ($this->Movie->saveAll($this->data, array('validate' => 'first'))) {
                $this->flash('Movie saved.', array('action'=>'index'));
// ...
```

The Attachments Element

What is it?

- A HTML only solution
- Form fields for attaching and detaching files
- A prototype to build your own solutions on (i.e. flash based)
- Uses the Helper internally

Usage

Modifying the template

app/views/movies/add.ctp

```
echo $form->create('Movie', array('type' => 'file'));
// ...
echo $this->element('attachments', array('plugin' => 'media'));
// ...
```

Advanced Demo

`http://localhost/workspace/media_demo/advanced/movies/add`

Future

- Keep up with new CakePHP releases
- Stabelize
- Few new features

This Is The End, My Friend

Questions?