File uploading, file storage and CakePHPs MediaView class

Written by Florian Kraemer on Mon, Jan 25th 2010, 14:36

This article includes how to upload and store files, because I've seen a lot of discussion about that too, but if you're just interested in how to use the MediaView class scroll down.

Handling file uploads in CakePHP

First let's start with the required form, to create a file upload form all you have to do is this:

Show Plain Text

```
echo $form->create('Media', array (http://www.php.net/array)('action' => 'upload', 'type' => 'file'));
echo $form->file (http://www.php.net/file)('file');
echo $form->submit(__('Upload', true));
```

The "type" in the options of Form::create() takes post, get or file. To configure the form for file uploading it has to be set to file which will render the form as a multipart/form-data form.

When you submit the form now, you'll get data like this in $this->data of your controller:

```
Array
(
        [Media] => Array
        (
                [file] => Array
                (
                        [name] => cake.jpg
                        [type] => image/jpeg
                        [tmp_name] => /tmp/hp1083.tmp
                        [error] => 0
                        [size] => 24530
                )
        )
)
```

Ok, now the big question with a simple answer is where the file data should be processed, guess where. Right – in the model because it's data to deal with and validation to do against it. Because it's a recurring task to upload files I suggest you to write a behaviour for it or convert your existing component to a behaviour.

If you keep it generic you can extend it with a CsvUpload, VideoUpload or ImageUpload behaviour to process the file directly after its upload or do special stuff with it, like resizing the image or parsing the csv file and store its data in a (associated) model.

We're not going to show you our own code here for obvious reasons, but I'll give you a few hints what you can or should do inside of the behavior:

1. Validate the uploaded field, the field itself contains already an error code if something was wrong with the upload. Here is a link to the php manual page that shows you the list of the errors that you can get from the form data. http://www.php.net/manual/en/features.file-upload.errors.php
2. Validate the uploaded file, is it really the kind of file you want and does it really contain the data structure you want?
3. Check if the target destination of the file is writeable, create directories, whatever is needed and error handling for it, I suggest you to use CakePHP's File and Folder classes for that.
4. Add a callback like beforeFileSave() and afterFileSave() to allow possible extending behaviors to use them.

Database vs file system storage

Feel free to skip that part if you already store the files in the file system.

Storing files in the database is in nearly all cases a bad solution because when you get the file it has to go its way through the database connection, which can, specially on servers that are not in the same network, cause performance problems.

Advantages of storage in the file system:

1. Easy and direct file access, to parse them (csv, xml...) or manipulate them (images)
2. You don't need to install any additional software to manage them
3. Easy to move and mount on other machines
4. Smaller then stored in a DB

The suggested solution is to store meta data of the file like size, hash, maybe path and other related info in a DB table and save the file in the file system.

Some people come up with the security and want to store a file because of that in the database which is wrong. You should not store the file in a public accessible directory like the webroot of the application. Store it in another location like APP/media. You control the access to the file by checking the permissions against the DB records of your meta data and sending it by using the CakePHP MediaView class, I'll explain later how to use it.

I don't say that storage of files inside the DB is in general a bad idea but for web based applications it is in nearly every case a bad idea.

File system Performance

A bottleneck in the long run on every file system is a large amount of files in a single directory. Imagine just 10.000 users and each has an individual avatar image. Further ext3 for example is limited to 32000 sub folders, other file systems have maybe similar restrictions. You can find a list of file system limitations here: http://en.wikipedia.org/wiki/Comparison_of_file_systems#Limits

To avoid performance problems caused by that you should store your files in a pseudo-random directory structure like APP/media/32/a5/3n/. This will also allow you to easily mount some of the semi-random created directories on another machine in the case you run out of disk space.

Show Plain Text

```
/**
 * Builds a semi random path based on the id to avoid having thousands of files
 * or directories in one directory. This would result in a slowdown on most file systems.
 *
 * Works up to 5 level deep
 *
 * @see http://en.wikipedia.org/wiki/Comparison_of_file_systems#Limits
 * @param mixed $string
 * @param integer $level
 * @return mixed
 * @access protected
 */
        protected function _randomPath($string, $level = 3) {
                if (!$string) {
                        throw new Exception(__('First argument is not a string!', true));
                }

                $string = crc32 (http://www.php.net/crc32)($string);
                $decrement = 0;
                $path = null;

                for ($i = 0; $i < $level; $i++) {
                        $decrement = $decrement -2;
                        $path .= sprintf (http://www.php.net/sprintf)("%02d" . DS, substr (http://www.php.net/su
bstr)('000000' . $string, $decrement, 2));
                }

                return $path;
        }
```

You should also know that php running in safe mode does not allow you to create more then one directory deep in one call. You have to take this in consideration, the above function does not cover that because safe mode is basically deprecated and will be also removed in php6

Sending a file to the client – or the unknown MediaView class

From what I've seen in the ruins of outsourced projects that asked us for rescue and also in the CakePHP googlegroup I think not many people are aware that CakePHP has a view that is thought to be used for downloads and display (images, text...) of files. It's called the MediaView class.

http://api.cakephp.org/class/media-view (http://api.cakephp.org/class/media-view)

I'll now explain you how to use this class to send files to the client.

Show Plain Text

```
/**
 * Sends a file to the client
 *
 * @param string $id UUID
 * @access public
 */
        public function download($id = null) {
                $this->Media->recursive = -1;
                $media = $this->Media->read(null, $id);

                if (empty (http://www.php.net/empty)($media)) {
                $this->redirect('/', 404, true);
                }

                $this->set('cache', '3 days');
                $this->set('download', true);
                $this->set('name', $media['Media']['slug']);
                $this->set('id', $media['Media']['filename']);
                $this->set('path', APP . 'media' . DS . $media['Media']['path']);
                $this->set('modified', $media['Media']['modified']);
                $this->set('mimeType', $media['Media']['mime_type']);
                $this->set('extension', $media['Media']['extension']);

                $this->view = 'Media';
                $this->autoLayout = false;
                if ($this->render() !== false) {
                        $this->Media->updateAll(
                                array (http://www.php.net/array)('Media.downloads' => 'Media.downloads + 1'),
                                array (http://www.php.net/array)('Media.id' => $id));
                }
        }
```

You simply have to set autoLayout to false and the view class to media.

Show Plain Text

```
$this->view = 'Media';
$this->autoLayout = false;
```

There are a few view variables to set to “configure” the file download or display. To control if you want to make the client downloading the file or to display it, in the case of images for example, you simply set 'download' to true or false;

Show Plain Text

```
$this->set('download', true);
```

You can control the browser caching of the file by setting cache. *Please not that you do not have to use caching if download is set to true! Downloads do not need caching.*

Show Plain Text

```
$this->set('cache', '3 days');
```

The next part might be a little confusing, you have “id” and “name”. Id is the actual file on your server you want to send while name is the filename under which you want to send the file to the client. “path” is the path to the file on the server.

Show Plain Text

```
$this->set('name', $media['Media']['slug']);
$this->set('id', $media['Media']['filename']);
$this->set('path', APP . 'media' . DS . $media['Media']['path']);
```

If you want to send a mime type that does not already in the MediaView class you can set it.

Show Plain Text

```
$this->set('mimeType', $media['Media']['mime_type']);
```

If you don't set it, the class will try to determine the mime type by the extension.

Show Plain Text

```
$this->set('extension', $media['Media']['extension']);
```

Note that you have to set the extension to make it work and that the extension is attached to the filename! If you store the filename with an extension you have to break it up.

When everything is set you can check if render() was successfully and do whatever you want after that, for example count the download.

Show Plain Text

```
if ($this->render() !== false) {
        $this->Media->updateAll(
        array (http://www.php.net/array)('Media.downloads' => 'Media.downloads + 1'),
        array (http://www.php.net/array)('Media.id' => $id));
}
```

Closing words

I hope you enjoyed reading the article and it helped you improving your knowledge about CakePHP. Feel free to ask further questions by using the comment functionality. Have fun coding!

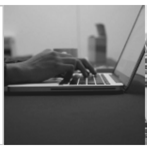
More Articles (/articles)

(http://www.addthis.com/bookmark.php?v=300&winname=addthis&pub=xa-52da11010a1b68cb&source=tbx32-300&lng=en-GB&s=google_plusone_share&url=http%3A%2F%2Fcakedc.com%2F&title=CakeDC%20-%20File%20uploading%2C%20file%20storage%20and%20CakePHPs%20MediaView%20class&ate=AT-xa-52da11010a1b68cb/-/-/532296bc78dabf17/2&frommenu=1&uid=532296bc278f2cb7&ct=1&pre=https%3A%2F%2Fwww.google.co.in%2F&tt=0&captcha_provider=nucaptcha)
(http://www.addthis.com/bookmark.php?v=300&winname=addthis&pub=xa-52da11010a1b68cb&source=tbx32-300&lng=en-GB&s=yahoobkm&url=http%3A%2F%2Fcakedc.com%2F&title=CakeDC%20-%20File%20uploading%2C%20file%20storage%20and%20CakePHPs%20MediaView%20class&ate=AT-xa-52da11010a1b68cb/-/-/532296bc78dabf17/3&frommenu=1&uid=532296bcbfd59f16&ct=1&pre=https%3A%2F%2Fwww.google.co.in%2F&tt=0&captcha_provider=nucaptcha)
(http://www.addthis.com/bookmark.php?v=300&winname=addthis&pub=xa-52da11010a1b68cb&source=tbx32-300&lng=en-GB&s=stumbleupon&url=http%3A%2F%2Fcakedc.com%2F&title=CakeDC%20-%20File%20uploading%2C%20file%20storage%20and%20CakePHPs%20MediaView%20class&ate=AT-xa-52da11010a1b68cb/-/-/532296bc78dabf17/4&frommenu=1&uid=532296bccdbcb93f&ct=1&pre=https%3A%2F%2Fwww.google.co.in%2F&tt=0&captcha_provider=nucaptcha)
(http://www.addthis.com/bookmark.php?v=300&winname=addthis&pub=xa-52da11010a1b68cb&source=tbx32-300&lng=en-GB&s=tumblr&url=http%3A%2F%2Fcakedc.com%2F&title=CakeDC%20-%20File%20uploading%2C%20file%20storage%20and%20CakePHPs%20MediaView%20class&ate=AT-xa-52da11010a1b68cb/-/-/532296bc78dabf17/5&frommenu=1&uid=532296bc4db8c494&ct=1&pre=https%3A%2F%2Fwww.google.co.in%2F&tt=0&captcha_provider=nucaptcha)
(http://www.addthis.com/bookmark.php?v=300&winname=addthis&pub=xa-52da11010a1b68cb&source=tbx32-300&lng=en-

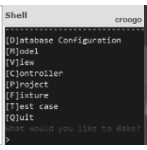
GB&s=dzone&url=http%3A%2F%2Fcakedc.com%2F&title=CakeDC%20-%20File%20uploading%2C%20file%20storage%20and%20CakePHPs%20MediaView%20class&ate=AT-xa-52da11010a1b68cb/-/-/532296bc78dabf17/6&frommenu=1&uid=532296bcb4154933&ct=1&pre=https%3A%2F%2Fwww.google.co.in%2F&tt=0&captcha_provider=nucaptcha)
(http://www.addthis.com/bookmark.php?v=300&winname=addthis&pub=xa-52da11010a1b68cb&source=tbx32-300&lng=en-GB&s=reddit&url=http%3A%2F%2Fcakedc.com%2F&title=CakeDC%20-%20File%20uploading%2C%20file%20storage%20and%20CakePHPs%20MediaView%20class&ate=AT-xa-52da11010a1b68cb/-/-/532296bc78dabf17/7&frommenu=1&uid=532296bc24a024cf&ct=1&pre=https%3A%2F%2Fwww.google.co.in%2F&tt=0&captcha_provider=nucaptcha)
(http://www.addthis.com/bookmark.php?v=300&winname=addthis&pub=xa-52da11010a1b68cb&source=tbx32-300&lng=en-GB&s=linkedin&url=http%3A%2F%2Fcakedc.com%2F&title=CakeDC%20-%20File%20uploading%2C%20file%20storage%20and%20CakePHPs%20MediaView%20class&ate=AT-xa-52da11010a1b68cb/-/-/532296bc78dabf17/8&frommenu=1&uid=532296bc72d6b232&ct=1&pre=https%3A%2F%2Fwww.google.co.in%2F&tt=0&captcha_provider=nucaptcha)

13

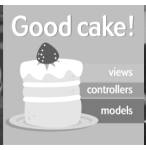
CakePHP



CakePHP CakePHP CakePHP CakePHP



```
Shell                                  croogo
[D]atabase Configuration
[M]odel
[V]iew
[C]ontroller
[P]roject
[F]ixture
[T]est case
[Q]uit
What would you like to Bake?
>
```



Services

- Custom Development (/development)
- Expert Consultancy (/consultancy)
- CakePHP Training (/training)

Community

- How We Help (/community)
- Plugin Standard (/plugin-standard)
- Git Workflow (/git-workflow)
- Articles (/articles)

Corporate

- About Us (/about)
- Our Team (/team)
- Employment (/jobs)
- Contact (/contact)

(http://www.w3.org/html/logo/)

CakePHP (http://cakephp.org)