CakePHP

Getting ready

Now that you have PHP installed let’s create a place in htdocs for your CakePHP development:

```
[tblgrant@silosilo htdocs]$ pwd
/u/tblgrant/apache/htdocs
[tblgrant@silosilo htdocs]$ mkdir cakewalks
[tblgrant@silosilo htdocs]$ cd cakewalks/
[tblgrant@silosilo cakewalks]$ pwd
/u/tblgrant/apache/htdocs/cakewalks
[tblgrant@silosilo cakewalks]$ ls
```

Downloading CakePHP

Make a folder one inside cakewalks and copy CakePHP in there:

```
[tblgrant@silosilo cakewalks]$ pwd
/u/tblgrant/apache/htdocs/cakewalks
[tblgrant@silosilo cakewalks]$ ls
[tblgrant@silosilo cakewalks]$ ls /l/www/classes/a202/software/cake*
/l/www/classes/a202/software/cakephp-cakephp-1.3.11-0-gc73ae84.tar.gz
[tblgrant@silosilo cakewalks]$ cp /l/www/classes/a202/software/cake* .
[tblgrant@silosilo cakewalks]$ ls
cakephp-cakephp-1.3.11-0-gc73ae84.tar.gz
[tblgrant@silosilo cakewalks]$ gunzip cakephp-cakephp-1.3.11-0-gc73ae84.tar.gz
[tblgrant@silosilo cakewalks]$ ls -l
-rw------- 1 tblgrant projects 983047 Jun 16 18:59 cakephp-cakephp-1.3.11-0-gc73ae84.tar
[tblgrant@silosilo cakewalks]$ ls
```
Uncompressing CakePHP

Already done in the previous section, via `gunzip`.

Installing CakePHP

Un-tar the archive, remove it. Rename the folder created and move in.

```
[tblgrant@silo cakewalks]$ pwd
/u/tblgrant/apache/htdocs/cakewalks
[tblgrant@silo cakewalks]$ ls
cakephp-cakephp-1.3.11-0-gc73ae84.tar
[tblgrant@silo cakewalks]$ tar xvf cakephp-cakephp-1.3.11-0-gc73ae84.tar
   cakephp-cakephp-3b830a4/
   cakephp-cakephp-3b830a4/.gitignore
   cakephp-cakephp-3b830a4/.htaccess
[...]
cakephp-cakephp-3b830a4/vendors/shells/templates/
   cakephp-cakephp-3b830a4/vendors/shells/templates/empty
[tblgrant@silo cakewalks]$ ls
cakephp-cakephp-1.3.11-0-gc73ae84.tar cakephp-cakephp-3b830a4
[tblgrant@silo cakewalks]$ rm *.tar
[tblgrant@silo cakewalks]$ ls
   cakephp-cakephp-3b830a4
[tblgrant@silo cakewalks]$ mv cakephp-cakephp-3b830a4/ one
[tblgrant@silo cakewalks]$ ls
   one
   [tblgrant@silo cakewalks]$ cd one
[tblgrant@silo one]$ pwd
/u/tblgrant/apache/htdocs/cakewalks/one
[tblgrant@silo one]$ ls
   app cake index.php plugins README vendors
[tblgrant@silo one]$ ls -a
   .. app cake .gitignore .htaccess index.php plugins README vendors
[tblgrant@silo one]$
```

Configuring CakePHP

We will follow the on-line manual\(^1\) at first. It doesn’t really matter in what order you choose to perform the steps below, as long as the steps are independent. The manual no longer says that but since our Apache is installed without mod_rewrite we need to configure CakePHP accordingly. This used to be one of the first steps, followed by the database configuration step. Let’s approach them in the same order below.

First, in your `~/apache/htdocs/cakewalks/one/app/config/core.php` file search for a section that looks like this:

\(^1\)\text{http://book.cakephp.org/1.3/view/876/The-Manual}
/*
 * To configure CakePHP *not* to use mod_rewrite and to
 * use CakePHP pretty URLs, remove these .htaccess
 * files:
 * *
 * /.htaccess
 * /app/.htaccess
 * /app/webroot/.htaccess
 * *
 * And uncomment the App.baseUrl below:
 */

//Configure::write('App.baseUrl', env('SCRIPT_NAME'));

Before you do anything access your CakePHP online as follows:

http://silo.cs.indiana.edu:61200/cakewalks/one/

Click on the index.php link that takes you to:


You need to take a look at this before you make any changes. Once you
saw the page please delete the three .htaccess files indicated in core.php and
uncomment the line below that comment. Here are the steps:

[tblgrant@silo config]$ pwd
/u/tblgrant/apache/htdocs/cakewalks/one/app/config
[tblgrant@silo config]$ pico -w core.php
[tblgrant@silo config]$ grep -i baseurl core.php
67: * And uncomment the App.baseUrl below:
69: Configure::write('App.baseUrl', env('SCRIPT_NAME'));
[tblgrant@silo config]$ ls -ld ../.htaccess
-rw------- 1 tblgrant projects 141 Jul 26 2011 ../.htaccess
[tblgrant@silo config]$ ls -ld ../../.htaccess
-rw------- 1 tblgrant projects 139 Jul 26 2011 ../../.htaccess
[tblgrant@silo config]$ ls -ld ../webroot/.htaccess
-rw------- 1 tblgrant projects 185 Jul 26 2011 ../webroot/.htaccess
[tblgrant@silo config]$ rm ../.htaccess
[tblgrant@silo config]$ rm ../../.htaccess
[tblgrant@silo config]$ rm ../webroot/.htaccess
[tblgrant@silo config]$ 

Now access the index.php again. Two things worth pointing out: (a)
the style sheets have kicked in and (b) the database configuration file is not
present. The instructions are to copy database.php.default into a new file
database.php and write inside the coordinates of your database server. Here’s
how I did that:
[tblgrant@silo config]$ pwd
/u/tblgrant/apache/htdocs/cakewalks/one/app/config
[tblgrant@silo config]$ ls
acl.ini.php bootstrap.php core.php database.php.default routes.php schema
[tblgrant@silo config]$ cp database.php.default database.php
[tblgrant@silo config]$ pico -w database.php
[tblgrant@silo config]$ diff database.php database.php.default
79,82c79,82
< 'host' => 'silo.cs.indiana.edu',
< 'login' => 'fabregas',
< 'password' => 'c3sc',
< 'database' => 'thursday',
---
> 'host' => 'localhost',
> 'login' => 'user',
> 'password' => 'password',
> 'database' => 'database_name',
84d83
< 'port' => '61230'
[tblgrant@silo config]$ 

As soon as I did that I accessed index.php online again. The page now says that the database configuration file is present and that Cake is able to connect to the database.
CakePHP Blog

Still following the online manual\(^2\) we will develop a simple Blog program. The steps that we are following are in chapter 11\(^3\) of the mentioned manual. Let’s review all the steps here.

We start with an introduction that enumerates the things we need\(^4\). We have or are about to acquire all of these. Next\(^5\) the tutorial instructs us to get and install Cake; we have already done that. Check that the info on the page is consistent with what you think (or, even better, remember) we have done thus far. The Blog application that we will develop must rely on a database, where all the posts will be stored. We create\(^6\) this database and because this is something we have not done yet we include the steps below:


<table>
<thead>
<tr>
<th>enrollment</th>
<th>matches</th>
<th>participation</th>
<th>players</th>
<th>spectators</th>
<th>venues</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

+-------------------+
|                |
+-------------------+

6 rows in set (0.00 sec)

mysql> /* First, create our posts table: */

mysql> CREATE TABLE posts (  
    -> id INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,  
    -> title VARCHAR(50),  
    -> body TEXT,  
    -> created DATETIME DEFAULT NULL,  
    -> modified DATETIME DEFAULT NULL  
    -> );

mysql> /* Then insert some posts for testing: */

mysql> INSERT INTO posts (title,body,created)  
    -> VALUES ('The title', 'This is the post body.', NOW());
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO posts (title,body,created)  
    -> VALUES ('A title once again', 'And the post body follows.', NOW());
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO posts (title,body,created)  
    -> VALUES ('Title strikes back', 'This is really exciting! Not.', NOW())
Query OK, 1 row affected (0.00 sec)

mysql> show tables;

<table>
<thead>
<tr>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Tables_in_thursday</td>
</tr>
</tbody>
</table>
+-------------------+
| enrollment         |
| matches            |
| participation      |
| players            |
| posts              |
| spectators         |
| venues             |
+-------------------+
7 rows in set (0.01 sec)

mysql> describe posts;

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Null</th>
<th>Default</th>
<th>Extra</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Next the tutorial instructs us to set up the database configuration\(^7\) and we've done that already. The next\(^8\) section is an optional configuration, which we will skip for now. It is followed by a note\(^9\) presenting considerations having to do with \texttt{mod_rewrite} and, again, this is something we have accomplished already\(^10\). From here on the tutorial actually starts the CakePHP development.

The tutorial goes step by step. We're going to take an orthogonal view to that and tell you up front that our development will result in:

- one model\(^{11}\)
- one controller\(^{12}\), and
- four views: index\(^{13}\), view\(^{14}\), add\(^{15}\) and edit\(^{16}\)

You know where to put them, let me show you the final contents of these four files:

\(^7\)\url{http://book.cakephp.org/1.3/en/view/1531/Cake-Database-Configuration}
\(^8\)\url{http://book.cakephp.org/1.3/en/view/1532/Optional-Configuration}
\(^9\)\url{http://book.cakephp.org/1.3/en/view/1533/A-Note-on-mod_rewrite}
\(^10\)However, it is a very important note and we will have to refer back to it one more time.
\(^11\)\url{app/models/post.php}
\(^12\)\url{app/controllers/posts_controller.php}
\(^13\)\url{app/views/posts/index.ctp}
\(^14\)\url{app/views/posts/view.ctp}
\(^15\)\url{app/views/posts/add.ctp}
\(^16\)\url{app/views/posts/edit.ctp}
<?php

class Post extends AppModel {
    var $name = 'Post';
    var $validate = array( 'title' => array( 'rule' => 'notEmpty' ),
        'body' => array( 'rule' => 'notEmpty' )
    );
}

?>

```php
<?php
class PostsController extends AppController {
    var $name = 'Posts';
    var $components = array('Session');

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

    function view($id) {
        $this->Post->id = $id;
        $this->set('post', $this->Post->read());
    }

    function add() {
        if (!empty($this->data)) {
            if ($this->Post->save($this->data)) {
                $this->Session->setFlash('Your post has been saved.');
                $this->redirect(array('action' => 'index'));
            }
        }
    }

    function edit($id = null) {
        if ($this->Post->id = $id; 
            if (empty($this->data)) {
                $this->data = $this->Post->read();
            } else {
                if ($this->Post->save($this->data)) {
                    $this->Session->setFlash('Your post has been updated.');
                    $this->redirect(array('action' => 'index'));
                }
            }
        }
    }

    function delete($id) {
        if ($this->Post->delete($id)) {
            $this->Session->setFlash('The post with id: ' . $id . ' has been deleted.');
            $this->redirect(array('action' => 'index'));
        }
    }
}
```
<h1>Blog posts</h1>
<p><?php echo $this->Html->link("Add Post", array('action' => 'add')); ?></p>
<table>
<tr>
<th>Id</th> 
<th>Title</th> <th>Action</th> <th>Created</th>
</tr>
<?php foreach ($posts as $post): ?>
<tr>
<td><?php echo $post['Post']['id']; ?></td>
<td><?php echo $this->Html->link($post['Post']['title'], array('action' => 'view', $post['Post']['id'])); ?></td>
<td><?php echo $this->Html->link('Delete', array('action' => 'delete', $post['Post']['id'], null, 'Are you sure?')); ?></td>
<td><?php echo $this->Html->link('Edit', array('action' => 'edit', $post['Post']['id'])); ?></td>
</tr>
<?php endforeach; ?></table>
The snapshot above shows the model, the controller and one of the views. Note that the folders for model(s), controller(s) already exist. We need to create the post folder inside the views folder before we start creating the .ctp files. As far the extension (.ctp) goes it’s supposed to stand for “cake[php] template”.

Here’s the source code for the other three template files (views):

```
[tblgrant@silo one]$ pwd
/u/tblgrant/apache/htdocs/cakewalks/one
[tblgrant@silo one]$ pico -w app/views/posts/edit.ctp
[tblgrant@silo one]$ cat app/views/posts/edit.ctp
<!-- File: /app/views/posts/edit.ctp -->
<h1>Edit Post</h1>

<?php echo $this->Form->create('Post', array('action' => 'edit'));
    echo $this->Form->input('title');
    echo $this->Form->input('body', array('rows' => '3'));
    echo $this->Form->input('id', array('type' => 'hidden'));
    echo $this->Form->end('Save Post');
?>
```

```
[tblgrant@silo one]$ pico -w app/views/posts/add.ctp
[tblgrant@silo one]$ cat app/views/posts/add.ctp
<!-- File: /app/views/posts/add.ctp -->
<h1>Add Post</h1>

<?php echo $this->Form->create('Post');
    echo $this->Form->input('title');
    echo $this->Form->input('body', array('rows' => '3'));
    echo $this->Form->end('Save Post');
?>
```

```
[tblgrant@silo one]$ pico -w app/views/posts/view.ctp
[tblgrant@silo one]$ cat app/views/posts/view.ctp
<!-- File: /app/views/posts/view.ctp -->
<h1><?php echo $post['Post']['title'] ?></h1>
<p><small>Created: <?php echo $post['Post']['created'] ?></small></p>
<p><?php echo $post['Post']['body'] ?></p>
```

Test your blog application with:

CakePHP 1.3 Recipes

This is a book we're making available under a special agreement. Ask me about getting access to it.