

WEBD 236

Web Information Systems Programming

Week 14

Copyright © 2012 Todd Whittaker
(todd.whittaker@franklin.edu)



Agenda

- This week's expected outcomes
- This week's topics
- This week's homework
- Upcoming deadlines
- Solution to Homework 11
- Questions and answers



Week 14 Outcomes

- List the advantages of using a web-framework for application development
- List and explain the typical features of a web-framework
- Examine some of the features of the Cake PHP framework.

Application Frameworks

- You may have noticed
 - Writing the same kind of code repeatedly
 - Models: getters, setters, findByX, insert, delete, update.
 - Controllers: retrieve parameters, validate, check permissions, update model, render a page or redirect
 - Views: using the same header/footer, printing variables
 - We have built a small application framework that takes some of the drudgery out

Application Frameworks

- Application Frameworks

- Provide the infrastructure for building apps, so you can concentrate on the problem

- Routing/dispatching
- Flexible MVC
- Caching
- Localization
- Validation/sanitization
- Security

All designed to work together, the foundations for any app. A production app framework provides much more.

Application Frameworks

- “Convention over configuration”

- Frameworks impose a way of doing things

- Ruby/Rails vs. Java/JEE

- Examples:

- Model class names are singular and camel case (e.g. BlogPost), but the table in the database is plural and underscored (e.g. blog_posts).
- Routes like /app/user/edit/5 get mapped to a class called UserController, calling the method edit(\$id), and 5 becomes the value of \$id.

Cake PHP

- Cake : a web framework for PHP
 - A “Rails for PHP” approach
 - Declarative and convention based rather than coding based.

A Blog in Cake

- Blogging example

Much of this comes from the “Blog Tutorial” on the Cake web site:
<http://book.cakephp.org/2.0/en/tutorials-and-examples/blog/blog.html>

A Blog in Cake

- DB config: app/config/database.php

```
<?php
class DATABASE_CONFIG {
    public $default = array(
        'datasource' => 'Database/Sqlite',
        'persistent' => false,
        'host' => 'localhost',
        'login' => 'user',
        'password' => 'password',
        'database' => 'blog.db3',
        'prefix' => ''
    );
}
```

Can also have a “test” configuration that is different. blog.db3 will appear in app/webroot

A Blog in Cake

- SQL for the posts table

```
CREATE TABLE posts (
    id INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,
    title VARCHAR(50),
    body TEXT,
    created DATETIME DEFAULT NULL,
    modified DATETIME DEFAULT NULL
)
INSERT INTO posts (title,body,created)
VALUES ('The title', 'This is the post body.', NOW());
INSERT INTO posts (title,body,created)
VALUES ('A title once again', 'And the post body follows.',
NOW());
INSERT INTO posts (title,body,created)
VALUES ('Title strikes back', 'This is really exciting!
Not.', NOW());
```

A Blog in Cake

- Post model in `app/model/Post.php`

```
<?php
class Post extends AppModel {
}
?>
```

Notice the model class is `Post`, but the table in the database is `posts`.

A Blog in Cake

- Post model in `app/model/Post.php`

```
<?php
class Post extends AppModel {
}
?>
```

`AppModel` is a class you can modify with application specific code. It extends `Model`, which provides many services (such as relationship mapping and “magic” finders, e.g. `findBy<fieldName>($value)`).

A Blog in Cake

- app/Controller/PostsController.php

```
<?php
class PostsController extends AppController {
    public $helpers = array('Html', 'Form');

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

This declaratively imports helper classes that can be used in the view. Can be put in AppController and then be available to all controllers automatically.

```
<?php
class PostsController extends AppController {
    public $helpers = array('Html', 'Form');

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

This method handles the URL app/posts/index.

A Blog in Cake

The associated view will have access to a posts variable.

A finder method defined in Model.

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

Every appropriately named controller will have a reference to the associated model.

A Blog in Cake

- View in app/View/Posts/index.ctp

```
<h1>Blog posts</h1>  
<table>  
  <tr><th>Id</th><th>Title</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('controller' => 'posts', 'action' => 'view',  
    $post['Post']['id'])); ?></td>  
    <td><?php echo $post['Post']['created']; ?></td>  
  </tr>  
  <?php endforeach; ?>  
</table>
```


A Blog in Cake

- View in app/View/Posts/index.ctp

```
<h1>Blog posts</h1>
<table>
  <tr><th>Id</th><th>Title</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('controller' => 'posts', 'action' => 'view',
        $post['Post']['id'])); ?></td>
    <td><?php echo $post['Post']['created']; ?></td>
  </tr>
  <?php endforeach; ?>
</table>
```

The name of the view is the same as the name of the method inside the controller.

A Blog in Cake

- View in app/View/Posts/index.ctp

```
<tr><th>Id</th><th>Title</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('controller' => 'posts', 'action' => 'view',
      $post['Post']['id'])); ?></td>
  <td><?php echo $post['Post']['created']; ?></td>
</tr>
<?php endforeach; ?>
</table>
```

This was "set" in the controller, and is available here.

The Html helper, link method. Links to posts/view/1.

A Blog in Cake

CakePHP: the rapid development php framework

Blog posts

Id	Title	Created
1	The title	2012-04-02 14:40:33
2	A title once again	2012-04-02 14:40:33
3	Title strikes back	2012-04-02 14:40:33
4	My post	2012-04-02 21:00:17

(default) 2 queries took 0 ms

Nr	Query	Error	Affected	Num. rows	Took (ms)
1	SELECT name FROM sqlite_master WHERE type='table' ORDER BY name;	0	0	0	0
2	SELECT "Post"."id", "Post"."title", "Post"."body", "Post"."created", "Post"."modified" FROM "main"."posts" AS "Post" WHERE 1 = 1	0	0	0	0

This
the
is av

title'],
ew',

A Blog in Cake

- app/Controller/PostsController

```
<?php
class PostsController extends AppController {
    public $helpers = array();

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

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

For the URL
posts/view/1.

A Blog in Cake

- View in app/View/Posts/view.ctp

```
<h1><?php echo h($post['Post']['title'])?></h1>

<p><small>Created: <?php echo $post['Post']['created']?>
</small></p>

<p><?php echo h($post['Post']['body'])?></p>
```

h() function is a wrapper for htmlspecialchars().

A Blog in Cake

- View

```
<h1><?php echo h($post['Post']['title'])?></h1>

<p><small>Created: <?php echo $post['Post']['created']?>
</small></p>

<p><?php echo h($post['Post']['body'])?></p>
```

The screenshot shows a web browser displaying a post view. The page content includes the title, creation date, and body text. Below the content is a debug console showing the execution of two SQL queries. The first query is a simple SELECT statement, and the second is a more complex query that joins the 'Post' table with the 'main.posts' table. The console also shows the number of rows affected and the execution time for each query.

Nr	Query	Error	Affected	Num. rows	Took (ms)
1	SELECT name FROM sqlite_master WHERE type='table' ORDER BY name;	0	0	0	1
2	SELECT "Post"."id", "Post"."title", "Post"."body", "Post"."created", "Post"."modified" FROM "main"."posts" AS "Post" WHERE "Post"."id" = 1 LIMIT 1	0	0	0	0

A Blog in Cake

- Move helpers up to ApplicationController:

```
class ApplicationController extends Controller {  
    public $components = array('Session');  
    public $helpers = array('Html', 'Form', 'Session');  
}
```

Sessions are needed to “flash” a status message to the user

A Blog in Cake

- Adding a post: the controller

```
// File: /app/Controller/PostsController.php  
  
public function add() {  
    if ($this -> request -> is('post')) {  
        if ($this -> Post -> save($this -> request -> data)) {  
            $this -> Session -> setFlash('Post has been saved.');            $this -> redirect(array('action' => 'index'));  
        } else {  
            $this -> Session -> setFlash('Unable to add post.');        }  
    }  
}
```

A Blog in Cake

- Always have access to the request w/o superglobal.

roller

Was it a post, put, get, or delete method?

```
// File: /app/controllers/postscontroller.php

public function add() {
    if ($this -> request -> is('post')) {
        if ($this -> Post -> save($this -> request -> data)) {
            $this -> Session -> setFlash('Post has been saved.');
```

Flash is accessible from session for status messages.

A Blog in Cake

- Adding a post: the view

```
<!-- File: /app/View/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');
?>
```

Form helper generates HTML for you.

A Blog in Cake

- Adding a post: the view

```
<!-- File: /app/View/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');
?>
```

Form helper generates HTML for you.
<form id="PostAddForm"
method="post" action="/posts/add">

A Blog in Cake

- Adding a post: validation in the model

```
// File: /app/Model/Post.php
class Post extends AppModel {
    public $validate = array(
        'title' => array(
            'rule' => 'notEmpty'
        ),
        'body' => array(
            'rule' => 'notEmpty'
        )
    );
}
```

This is declarative validation. The view will inspect the model to know how to present it.

A Blog in Cake

The screenshot shows a web browser window with the title "CakePHP: the rapid development php framework". The page content is titled "Add Post". There is a form with two fields: "Title*" and "Body*". The "Title*" field contains the text "A post with an empty body". The "Body*" field is empty. Below the form is a green button labeled "Save Post". At the bottom of the page, there is a status bar that says "(default) 0 query took ms". Below the status bar is a table with the following columns: "Nr", "Query", "Error", "Affected", "Num. rows", and "Took (ms)".

On the left side of the browser window, there is a code editor showing the following code:

```
//  
cl  
}
```

On the right side of the browser window, there is a blue rounded rectangle containing the text:

ive
e view
e model
o

A Blog in Cake

The screenshot shows a web browser window with the title "CakePHP: the rapid development php framework". The page content is titled "Add Post". There is a form with two fields: "Title*" and "Body*". The "Title*" field contains the text "A post with an empty body". The "Body*" field is empty. Below the form is a green button labeled "Save Post". A red error message box is displayed at the top of the form, containing the text "Unable to add your post.". Below the error message, the form fields are visible again. The "Body*" field is highlighted in orange, and a message below it says "This field cannot be left blank". Below the form is a green button labeled "Save Post".

On the left side of the browser window, there is a code editor showing the following code:

```
//  
cl  
}
```

A Blog in Cake

CakePHP: the rapid development php framework

Add Post

Title*

A post with

Body*

Save Post

Unable to add your post

Add Post

Title*

A post with an empty body

Body*

This field cannot be left blank

Save Post

No code for:

- Retaining values
- Flashing the message
- Displaying errors

```
//  
cl  
  
}  
Nr Query
```

A Blog in Cake

- Editing a post: the controller

```
// File: /app/Controller/PostsController.php
```

```
public function edit($id = null) {  
    $this->Post->id = $id;  
    if ($this->request->is('get')) {  
        $this->request->data = $this->Post->read();  
    } else {  
        if ($this->Post->save($this->request->data)) {  
            $this->Session->setFlash('Post updated.');            $this->redirect(array('action' => 'index'));  
        } else {  
            $this->Session->setFlash('Unable to update post.');        }  
    }  
}
```


A Blog in Cake

- Editing a post: the view

```
<!-- File: /app/View/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');
?>
```

A Blog in Cake

- Ed

```
<!-- F
<h1>Ed
<?php
echo $
echo $
echo $
echo $
echo $
?>
```

CakePHP: the rapid development php framework

Edit Post

Title*

Body*

CAKEPHP POWER

(default) 1 query took 0 ms

Nr Query	Error Affected	Num. rows	Took (ms)

A Blog in Cake

- Adding links: app/View/Posts/index.ctp

```

<!-- ...stuff skipped -->
<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('controller' => 'posts', 'action' => 'view',
        $post['Post']['id'])); ?></td>
<td><?php echo $this->Html->link('Edit', array('action' =>
    'edit', $post['Post']['id'])); ?></td>
<td><?php echo $post['Post']['created']; ?></td>
</tr>
<?php endforeach; ?>
    
```

A Blog in Cake

- Add Post

Blog posts

[Add Post](#)

Id	Title	Action	Created
1	The title	Edit	2012-04-02 14:40:33
2	A title once again	Edit	2012-04-02 14:40:33
3	Title strikes back	Edit	2012-04-02 14:40:33
4	My post	Edit	2012-04-02 21:00:17

(default) 2 queries took 0 ms

Nr	Query	Error	Affected	Num. rows	Took (ms)
1	SELECT name FROM sqlite_master WHERE type='table' ORDER BY name;	0	0	0	0
2	SELECT "Post"."id", "Post"."title", "Post"."body", "Post"."created", "Post"."modified" FROM "main"."posts" AS "Post" WHERE 1 = 1	0	0	0	0

A Blog in Cake

- Deleting posts: controller

```
// File: /app/Controller/PostsController.php

public function delete($id) {
    if ($this -> request -> is('get')) {
        throw new MethodNotAllowedException();
    }
    if ($this -> Post -> delete($id)) {
        $this -> Session -> setFlash('The post with id: '
            . $id . ' has been deleted. ');
        $this -> redirect(array('action' => 'index'));
    }
}
```

A Blog in Cake

- Deleting posts: controller

```
// File: /app/Controller/PostsController.php

public function delete($id) {
    if ($this -> request -> is('get')) {
        throw new MethodNotAllowedException();
    }
    if ($this -> Post -> delete($id)) {
        $this -> Session -> setFlash('The post with id: '
            . $id . ' has been deleted. ');
        $this -> redirect(array('action' => 'index'));
    }
}
```

Preventing a get-request from deleting prevents web-spiders from wiping out your database.

A Blog in Cake

- Adding links: app/View/Posts/index.ctp

```
<!-- ...stuff skipped -->
<td><?php echo $this->Html->link($post['Post']['title'],
    array('controller' => 'posts', 'action' => 'view',
    $post['Post']['id'])); ?></td>
<td>
    <?php echo $this->Html->link('Edit', array('action' =>
    'edit', $post['Post']['id'])); ?>
    <?php echo $this->Form->postLink('Delete',
    array('action' => 'delete', $post['Post']['id']),
    array('confirm' => 'Are you sure?')); ?>
</td>
<td><?php echo $post['Post']['created']; ?></td>
</tr>
<?php endforeach; ?>
```

A Blog in Cake

- Add

The screenshot shows a web browser displaying a list of blog posts. The page title is "Blog posts" and there is a link for "Add Post". The posts are listed in a table with columns for Id, Title, Action, and Created. Below the table, there is a status bar indicating "(default) 2 queries took 0 ms" and a query log table.

Id	Title	Action	Created
1	The title	Edit Delete	2012-04-02 14:40:33
2	A title once again	Edit Delete	2012-04-02 14:40:33
3	Title strikes back	Edit Delete	2012-04-02 14:40:33
4	My post	Edit Delete	2012-04-02 21:00:17

Nr	Query	Error	Affected	Num. rows	Took (ms)
1	SELECT name FROM sqlite_master WHERE type='table' ORDER BY name;	0	0	0	0
2	SELECT "Post"."id", "Post"."title", "Post"."body", "Post"."created", "Post"."modified" FROM "main"."posts" AS "Post" WHERE 1 = 1	0	0	0	0

ctp

```
<!-- ...stuff skipped -->
<td><?php echo $this->Html->link($post['Post']['title'],
    array('controller' => 'posts', 'action' => 'view',
    $post['Post']['id'])); ?></td>
<td>
    <?php echo $this->Html->link('Edit', array('action' =>
    'edit', $post['Post']['id'])); ?>
    <?php echo $this->Form->postLink('Delete',
    array('action' => 'delete', $post['Post']['id']),
    array('confirm' => 'Are you sure?')); ?>
</td>
<td><?php echo $post['Post']['created']; ?></td>
</tr>
<?php endforeach; ?>
```

A Blog in Cake

- Add

CakePHP: the rapid development php framework

Blog posts

Add Post

Id	Title		
1	The title		
2	A title once again		
3	Title strikes back		
4	My post	Edit Delete	2012-04-02 21:00:17

The page at localhost says:
Are you sure?
OK Cancel

(default) 2 queries took 0 ms

Nr	Query	Error	Affected	Num. rows	Took (ms)
1	SELECT name FROM sqlite_master WHERE type='table' ORDER BY name;	0	0	0	0
2	SELECT "Post"."id", "Post"."title", "Post"."body", "Post"."created", "Post"."modified" FROM "main"."posts" AS "Post" WHERE 1 = 1	0	0	0	0

```
<!-- .
<t
<t
</tr>
<?php endforeach; ?>
```

A Blog in Cake

- Add

CakePHP: the rapid development php framework

The post with id: 4 has been deleted.

Blog posts

Add Post

Id	Title	Action	Created
1	The title	Edit Delete	2012-04-02 14:40:33
2	A title once again	Edit Delete	2012-04-02 14:40:33
3	Title strikes back	Edit Delete	2012-04-02 14:40:33

(default) 1 query took 0 ms

Nr	Query	Error	Affected	Num. rows	Took (ms)
1	SELECT "Post"."id", "Post"."title", "Post"."body", "Post"."created", "Post"."modified" FROM "main"."posts" AS "Post" WHERE 1 = 1	0	0	0	0

```
</tr>
<?php endforeach; ?>
```

A Blog in Cake

- Routing
 - Connecting URLs to controllers
 - Right now, blogs are at /app/posts. We'd like to redefine that to be /app.
 - Edit app/Config/routes.php

```
Router::connect(  
    '/',  
    array(  
        'controller' => 'posts',  
        'action' => 'index'  
    )  
);
```

A Blog in Cake

- Authorization
 - Follow the same kinds of procedure to create
 - a users table in the database,
 - the model in app/Model/User.php,
 - The controller in app/Controller/UsersController.php (with CRUD methods)
 - the needed templates in app/View/Users (CRU)

A Blog in Cake

- Authorization

- Follow the same kinds of procedure to create

- a users table in the database,
- the model in `app/Model/User.php`,
- The controller in `app/Controller/UsersController.php` (with CRUD, login, logout methods)
- the needed templates in `app/View/Users` (CRU)
- Add a `user_id` field to posts DB table

A Blog in Cake

- Authorization

```
class AppController extends Controller {
    public $helpers = array('Html', 'Form', 'Session');
    public $components = array(
        'Session',
        'Auth' => array(
            'loginRedirect' => array('controller' => 'posts',
                'action' => 'index'),
            'logoutRedirect' => array('controller' => 'posts',
                'action' => 'index'),
            'authorize' => array('Controller')
        )
    );
    // ... more ...
}
```

A Blog in Cake

- Authorization

Pre-existing Auth component.
Configuration, not code.

```
class AppController extends Controller {
    public $helpers = array('Form', 'Session');
    public $components = array(
        'Session',
        'Auth' => array(
            'loginRedirect' => array('controller' => 'posts',
                'action' => 'index'),
            'logoutRedirect' => array('controller' => 'posts',
                'action' => 'index'),
            'authorize' => array('Controller')
        )
    );
    // ... more ...
}
```

A Blog in Cake

- Authorization

```
class AppController extends Controller {

    public function beforeFilter() {
        $this->Auth->allow('index', 'view');
    }

    public function isAuthorized($user) {
        // Admin can access every action
        if (isset($user['role']) && $user['role'] === 'admin') {
            return true;
        }
        // Default deny
        return false;
    }

}
```


A Blog in Cake

- Authorization

Allow index and view methods without authorization or login.

```
class AppController extends Controller {

    public function beforeFilter() {
        $this->Auth->allow('index', 'view');
    }

    public function isAuthorized($user) {
        // Admin can access every action
        if (isset($user['role']) && $user['role'] === 'admin') {
            return true;
        }
        // Default deny
        return false;
    }
}
```

UNIVERSITY

A Blog in Cake

- Authorization: in PostsController.php

```
public function isAuthorized($user) {
    // All registered users can add posts
    if ($this -> action === 'add') {
        return true;
    }
    // The owner of a post can edit and delete it
    if (in_array($this -> action, array('edit','delete'))) {
        $postId = $this -> request -> params['pass'][0];
        if ($this -> Post -> isOwnedBy($postId,
            $user['id'])) {
            return true;
        }
    }
    return parent::isAuthorized($user);
}
```

UNIVERSITY

A Blog in Cake

- Authorization: in model Post.php

```
public function isOwnedBy($post, $user) {  
    return $this -> field('id', array('id' => $post,  
        'user_id' => $user)) === $post;  
}
```

Many more features

- Additional features:
 - ACL based authorization
 - Object relational mapping of one-to-many (hasMany) and many-to-many (hasAndBelongsToMany) relationships
 - Code generation from an existing DB (bake)
 - Email components
 - Pagination, cookies, sessions, security, etc.

Learning curve

- Learning a framework is hard
 - Resources
 - Online books and tutorials:
<http://book.cakephp.org/2.0/en/>
 - Free ebooks: <http://0-ebooks.ohiolink.edu.olinserver.franklin.edu/ebooks/home/>
 - The payoff in development time is huge.

Other frameworks

- PHP has many frameworks
 - E.g. Yii, CodeIgniter, Zend, Symfony
 - Each has a different philosophy (monolithic vs. plugin-based) and provides different features.
- And several templating languages
 - E.g. Smarty, Dwoo, Rain

Solution to HW 11

Next Week

- Final Exam!

General Q & A

- Questions?
- Comments?
- Concerns?