



- WIFI:
 - user: special
 - password: group
- Pick randomly from
 - meeting 3,4,5,7

CakeFest 2013

San Francisco, USA

CakePHP Basic Workshop: Intro - Welcome!

- About CakePHP
 - MVC Framework
 - MIT License
 - 7+ years (mature)
 - Convention over configuration
 - Order and structure
 - Plugins / Testing
 - Strong community
 - <http://community.cakephp.org>
- About me
 - CakePHP developer @ CakeDC
 - Basic CakePHP Training Course @ CSF



Development environment: Setup

- Apache 2.2.x
- MySQL 5
- PHP 5.3 (>5.2.8)
- CakePHP latest master
- Git

XAMPP / LAMPP (quick way)

```
$ sudo tar xvfz xampp-linux-1.7.7.tar.gz -C /opt  
$ sudo /opt/lampp/lampp start
```



Development environment: Creating a project

- Shared Core or Core per project?
- Download CakePHP latest and extract OR clone into
 - /usr/share/cakephp/latest
- Link your shared core
 - `ln -s /usr/share/cakephp/latest/lib .`
- Bake your new project, let's call it "jobadvisor"

```
~/workshop $ git clone  
https://github.com/steinkel/cakefest-basic-workshop-2013.git jobadvisor
```

```
~/workshop $ cd jobadvisor
```

```
~/workshop $ git reset --hard start
```

```
~/workshop $ cd ..
```

```
~/workshop $ lib/Cake/Console/cake bake project jobadvisor --empty
```

Development environment: Folder Structure

- Root (~/workshop)
 - /lib → /usr/share/cakephp/2.3/lib
 - /jobadvisor
 - /Config
 - /Controller
 - /Model
 - /View
 - /webroot
 - ...
 - /project2
 - /project3
 - ...

Apache virtual host config

job.com →

~/workshop/jobadvisor/webroot



Continue?



CakePHP: the rapid development php framework

Release Notes for CakePHP 2.3.9.

[Read the changelog](#)

Your version of PHP is 5.2.8 or higher.

Your tmp directory is writable.

The *FileEngine* is being used for core caching. To change the config edit APP/Config/core.php

Your database configuration file is NOT present.
Rename APP/Config/database.php.default to APP/Config/database.php

DebugKit is not installed. It will help you inspect and debug different aspects of your application.
You can install it from [github](#)



Non standard setups

Fix paths using webroot/index.php relevant constants

webroot

< Point here your webserver document root

core

< Use `CAKE_CORE_INCLUDE_PATH`
Point the 'lib' folder, parent to the Core files

app

< Use `ROOT` and `APP_DIR`
`ROOT` → Parent folder of all the apps
`APP_DIR` → Name of the app dir, usually "app"



Non standard setups : example

```
~/custom $ ls  
app/  
lib@  
webroot/
```

Constants used

```
define('ROOT', dirname(dirname(__FILE__)));
```

```
define('APP_DIR', 'app');
```

```
define('CAKE_CORE_INCLUDE_PATH', ROOT . DS . 'lib');
```



Configure database

```
$ wget https://dl.dropboxusercontent.com/u/142951/dump_jobadvisor.sql  
$ echo 'CREATE DATABASE jobadvisor;' | mysql  
$ echo 'CREATE DATABASE jobadvisor_test;' | mysql  
$ mysql jobadvisor < dump_jobadvisor.sql
```

Configure your database.php file



Setting up DebugKit Plugin

(if not using the workshop repo) \$ git init

\$ git submodule add https://github.com/cakephp/debug_kit.git Plugin/DebugKit

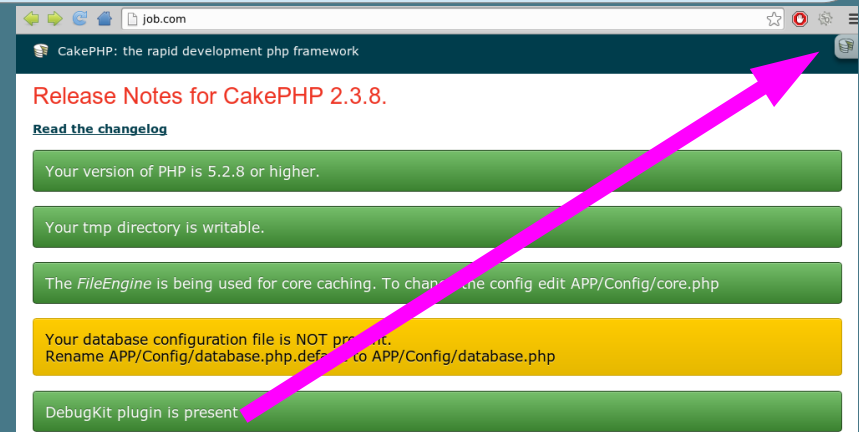
- * Download / Submodule

- * Load in Bootstrap

```
CakePlugin::load('DebugKit');
```

- * Load Component in AppController

```
public $components = array('Session', 'DebugKit.Toolbar');
```

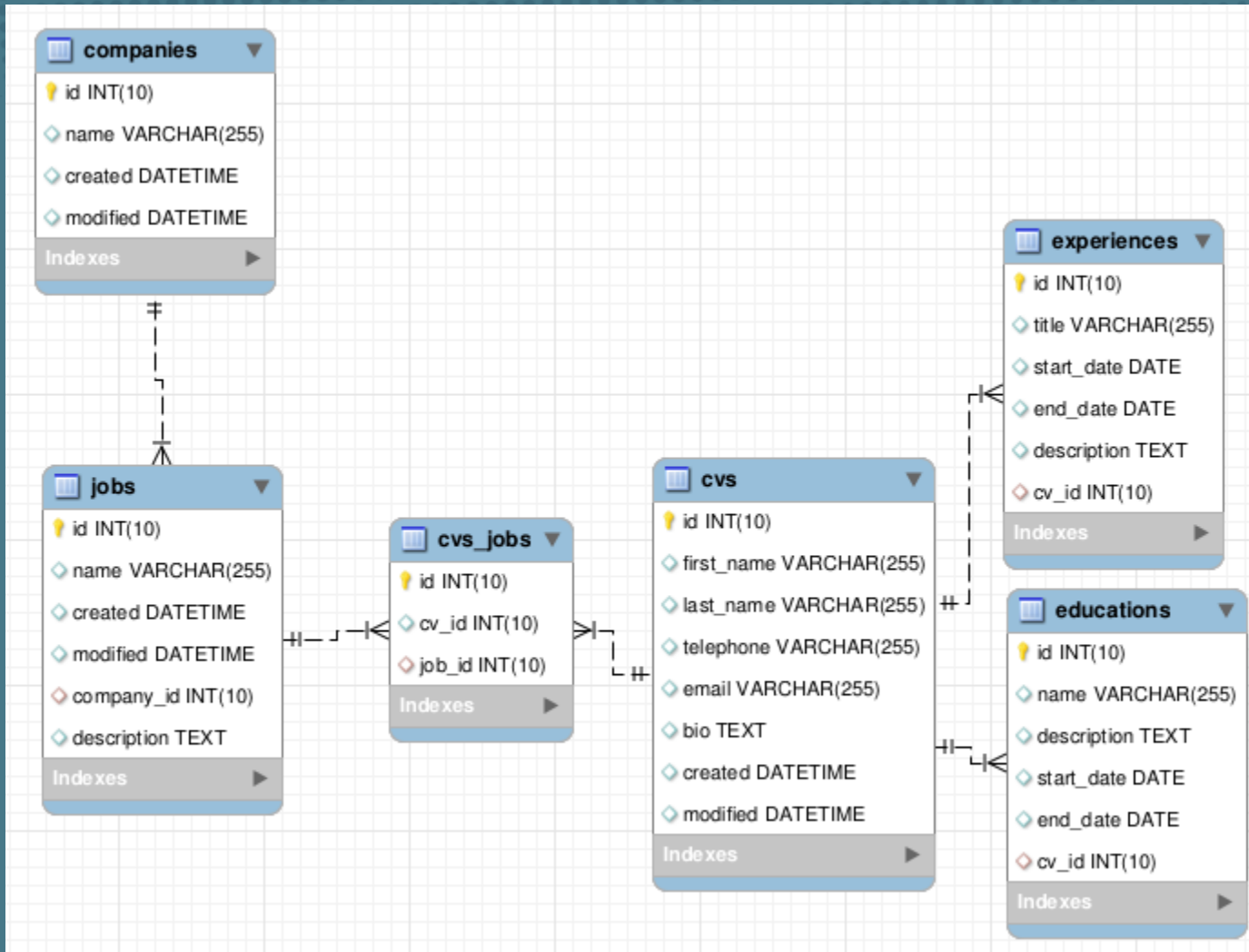


JobAdvisor, our sample app (intro)

- Job Advisor Beta*
 - Users may create a CV and apply for Jobs
 - Companies may create and manage Jobs
 - Public frontend for latest Jobs and Companies
- We'll build:
 - Public backend for CVs, Companies and Jobs
 - Home page, listing latest jobs and companies
 - Detail pages for jobs and companies



JobAdvisor, DB schema (initial)



Building a default CRUD (non interactive mode)

```
$ Console/cake bake model all  
$ Console/cake bake controller all  
$ Console/cake bake view all
```

* Basic code generation for quick CRUDs

The screenshot displays a web application interface for managing CVs. On the left, a sidebar titled 'Actions' contains buttons for 'List Cvs', 'List Educations', 'New Education', 'List Experiences', 'New Experience', 'List Jobs', and 'New Job'. The main content area is titled 'Add Cv' and features four input fields: 'First Name', 'Last Name', 'Telephone', and 'Email'. To the right of the form, there is a table with the following data:

	Company	Description	Actions
24	ACME Ltd	Awesome job	View Edit Delete
1			

Below the table, it states 'total, starting on record 1, ending on 1'. The application's header and footer both indicate 'CakePHP: the rapid development php framework'.



CakePHP Models: Introduction

- Business logic
- Validation
- Associations
- Usually linked to a database table
- Examples
 - Job (jobs)
 - CV (cvs)
 - Company (companies) ← note inflection



CakePHP Models: Definition and properties

- extends AppModel
 - useTable (default name pluralized lowercase)
 - primaryKey (default 'id')
 - displayField (default 'name' or 'title')
 - recursive (default 0)
 - validate
 - virtualFields
 - defined model associations

```
<?php
App::uses('AppModel', 'Model');
/**
 * Company Model
 *
 * @property Job $Job
 */
class Company extends AppModel {
    public $hasMany = array('Job');
}
```



CakePHP Models: Setting a “complex” display field for Cv

- Company Model

```
public $displayField = 'display_name';  
public $virtualFields = array(  
    'display_name' => "CONCAT(Company.name, ' - ', Company.id)"  
);
```

- Or use the constructor

```
public function __construct($id = false, $table = null, $ds = null) {  
    parent::__construct($id, $table, $ds);  
    $this->virtualFields = array(  
        'display_name' => "CONCAT({$this->alias}.name, ' - ', {$this->alias}.{$this->primaryKey})",  
    );  
}
```

- Do the same with the Cv model



CakePHP Models: Callbacks

- `beforeFind(array $queryData)`
- `afterFind(array $results, boolean $primary = false)`
- (*) `beforeValidate(array $options = array())`
- (*) `beforeSave(array $options = array())`
- `afterSave(boolean $created)`
- (*) `beforeDelete(boolean $cascade = true)`
- `afterDelete()`
- `onError()`



CakePHP Models: Callbacks - email me on new Companies

- Configure email.php
- afterSave(boolean \$created)

```
public function afterSave(boolean $created) {  
    if ($created) {  
        CakeEmail::deliver(  
            'admin@job.com',  
            __("New Company created id: %s", $this->data['Company']['id']),  
            __('New company was created ...'),  
            'default'  
        );  
    }  
}
```

- Bad practice alert!! Do it right:
 - Using a Process queue
 - <http://www.slideshare.net/mgiglesias/job-queues-with-gearman>



CakePHP Models: Associations - Intro and properties

- Job belongsTo Company

```
public $belongsTo = array(
    'Company' => array(
        'className' => 'Company',
        'foreignKey' => 'company_id',
        'conditions' => '',
        'fields' => '',
        'order' => ''
    )
);
```

- Cv hasMany Education

```
public $hasMany = array(
    'Education' => array(
        'className' => 'Education',
        'foreignKey' => 'cv_id',
        'dependent' => false,
        'conditions' => '',
        'fields' => '',
        'order' => '',
        'limit' => '',
        'offset' => '',
        'exclusive' => '',
        'finderQuery' => '',
        'counterQuery' => ''
    ),
);
```

- Cv hasAndBelongsToMany Job

```
public $hasAndBelongsToMany = array(
    'Job' => array(
        'className' => 'Job',
        'joinTable' => 'cvs_jobs',
        'foreignKey' => 'cv_id',
        'associationForeignKey' => 'job_id',
        'unique' => 'keepExisting',
        'conditions' => '',
        'fields' => '',
        'order' => '',
        'limit' => '',
        'offset' => '',
        'finderQuery' => '',
        'deleteQuery' => '',
        'insertQuery' => ''
    )
);
```

- Add an 'UnfinishedPositions' association to the Cv model



CakePHP Models: Validation - Properties and basic rules

- Simple, complex and custom declarative validation per field
 - field → array
 - name → array
 - rule
 - message
 - required / allowEmpty
 - on / last

- comparison / equalTo
- inList / maxLenght / range
- isUnique / notEmpty
- decimal / numeric / naturalNumber
- and more ...

```
// adding some validation rules
public $validate = array(
    'name' => 'alphaNumeric',
    'description' => array(
        'alphaNumeric' => array(
            'rule' => 'alphaNumeric',
            'message' => 'Please, use only characters and numbers in description'
        ),
        'minLength' => array(
            'rule' => array('minLength', '8'),
            'message' => 'Description should have at least 8 characters'
        )
    )
);
```



CakePHP Models: Validation – adding your custom rules

- Limit creation of new Jobs in Weekdays

```
public $validate = array(  
    'name' => array(  
        'todayIsWorkday' => array(  
            'rule' => array('todayIsWorkday'),  
            'message' => 'You can\'t create new jobs in weekends',  
            'on' => 'create',  
            'last' => true,  
        ),  
    ),  
);
```



CakePHP Models: Reading data

- `read($fields, $id)`
- `find('type', $options)`
 - 'first'
 - 'count'
 - 'all'
 - 'list'
 - 'threaded' / 'neighbors'
 - 'custom' finders
- `field('field', $conditions)`
- Magic finders

```
$jobs = $this->find('all');
```

```
$jobs = $this->findAllByName('test');
```

conditions

recursive

fields

order

group

limit, page, offset

callbacks



CakePHP Models: Reading data – Result array

```
/**
 * Sample function to get related jobs to a companyId
 * @param int $companyId Company ID
 * @return array related jobs
 */
public function getRelatedJobs($companyId) {
    return $this->Job->find('all', array(
        'conditions' => array('Job.company_id' => $companyId),
        'order' => array('Job.created DESC')
    ));
}
```

```
(
  1 [0] => Array
    (
      2 [Job] => Array
        (
          [id] => 3
          [name] => PHP Developer
          [created] => 2012-10-01 22:23:15
          [modified] => 2012-10-01 22:23:15
          [company_id] => 1
        )

      3 [Company] => Array
        (
          [id] => 1
          [name] => ACME Consulting
          [created] => 2012-10-01 22:22:14
          [modified] => 2012-10-01 22:22:14
          [nameId] => ACME Consulting - 1
        )

      [Cv] => Array
        (
        )
    )
)
```



CakePHP Models: Reading data – using custom finders

- Latest Jobs

```
public $findMethods = array('latest' => true);

protected function _findLatest($state, $query, $results = array()) {
    if ($state == 'before') {
        ....
        return $query;
    }
    return $results;
}
```



CakePHP Models: Reading data – using on the fly associations

- Completed Experiences, added to the view action

```
$this->Cv->bindModel(
    array('hasMany' => array(
        'CompletedExperience' => array(
            'className' => 'Experience',
            'conditions' => array('end_date < NOW()'),
        )
    )
);
```



CakePHP Models: Saving data

- create()
- save(\$data, \$options)
- saveField('field', \$value)
- saveMany(\$data)
- saveAssociated(\$data)
- saveAll(\$data)

```
debug($this->Cv->saveField('bio', 'Updated bio'));
```

```
array(
    'Cv' => array(
        'id' => '15',
        'bio' => 'Updated bio',
        'modified' => '2012-10-03 13:05:59'
    )
)
```

```
/**
 * doing sample CV adding
 */
public function saving_samples() {
    $data = array(
        'first_name' => 'Peter',
        'last_name' => 'Parker',
        'telephone' => '666777888',
        'email' => 'pparker@example.com',
        'bio' => 'His bio here...'
    );
    $this->Cv->create();
    $saveResult = $this->Cv->save($data);
    debug($saveResult);
    $this->_stop();
}
```



CakePHP Models: Saving data – complex structures

- saveAssociated(\$data)
- saveAll tip: deeper associations

```
$complexData = array(  
    'Cv' => array(  
        'first_name' => 'SaveAll',  
        'last_name' => 'Testing',  
        'telephone' => '666777888',  
        'email' => 'test@example.com',  
        'bio' => 'Another bio...'  
    ),  
    'Education' => array(  
        array('name' => 'first', 'description' => 'first description'),  
        array('name' => 'second', 'description' => 'second description'),  
    )  
);
```



CakePHP Behaviours: Reusing model logic

- Refactor our “send email on create” callback into a new Behavior

```
App::uses('ModelBehavior', 'Model');
App::uses('CakeEmail', 'Network/Email');
class CreationReportBehavior extends ModelBehavior {
    public function afterSave(Model $Model, $created) {
        if ($created) {
            $mail = CakeEmail::deliver(
                'admin@job.com',
                __("New %s created id: %s",
                    $Model->alias,
                    $Model->data[$Model->alias]['id']),
                __('New %s was created ...',
                    $Model->alias),
                'default');
        }
    }
}
```



CakePHP Models: Using Containable Behavior

- Recursive is no more!
- Select exactly what you need
- get companies and job names for all jobs with description like "PHP"
- get CV names for Jobs in a company

```
$this->Company->find('all', array(  
    'contain' => array('Job' => array(  
        'conditions' => array("Job.description LIKE '%PHP%'"),  
        'fields' => array('Job.name'),  
    ))  
))
```



CakePHP Controllers: Introduction

- About AppController
- Using models
 - \$uses
- Affecting the views
 - set()
 - render()
- Flow control
 - redirect()
- Controller Callbacks:
 - beforeFilter()
 - beforeRender()
 - afterFilter()
- Using Components and Helpers



CakePHP Request

- Accessing the request
 - `$this->request->params`
 - `$this->request->query('value')`
 - `$this->request->data('value')`
- `$this->request->is('detector')`
 - 'post' / 'get' / 'ajax' / 'mobile' / 'ssl'

```
object(CakeRequest) {  
    params => array(  
        'plugin' => null,  
        'controller' => 'jobs',  
        'action' => 'hello',  
        'named' => array(),  
        'pass' => array()  
    )  
    data => array()  
    query => array()  
    url => 'jobs/hello'  
    base => '/jobadvisor/'  
    webroot => '/jobadvisor/'  
    here => '/jobadvisor/jobs/hello'  
    [protected] _detectors => array(  
        'post' => true,  
        'get' => true,  
        'ajax' => true,  
        'mobile' => true,  
        'ssl' => true  
    )  
}
```



CakePHP Response

- Modify the response sent
 - `$this->response->statusCode($c)`
 - `$this->response->body($content)`
 - `$this->response->header()`
 - `$this->response->type($type)`
 - `$this->response->expires($time)`
 - `$this->response->disableCache();`
- Send response
 - `$this->response->send()`

- `$this->response->file(
 $path,
 array(
 'download' => true,
 'name' => $name));`

- Downloading the error log



Break



CakePHP Views: Introduction

- Layouts
 - Views
 - Elements
 - Blocks
 -
- CakePHP default layout
-
- Helpers
 - Using helpers in views



CakePHP Views: Html Helper

- Available by default in all views
 - Generates well formatted html element also can save you some coding
 -
 - \$options array
 - array('class' => 'short-label')
 - becomes class = 'short-label' attribute
- image(\$path, \$options)
 - link(\$title, \$url, \$options)
 - url(\$url, \$full) - wraps Router::url()
 -
 - ADD table examples



CakePHP Views: Form Helper

- tbd

- `image($path, $options)`
- `link($title, $url, $options)`
- `url($url, $full)` - wraps `Router::url()`
-
- ADD table examples



CakePHP Views: Other Core Helpers

- tbd

- image(\$path, \$options)
- link(\$title, \$url, \$options)
- url(\$url, \$full) - wraps Router::url()
-
- ADD table examples



CakePHP Helpers: Creating a new Helper

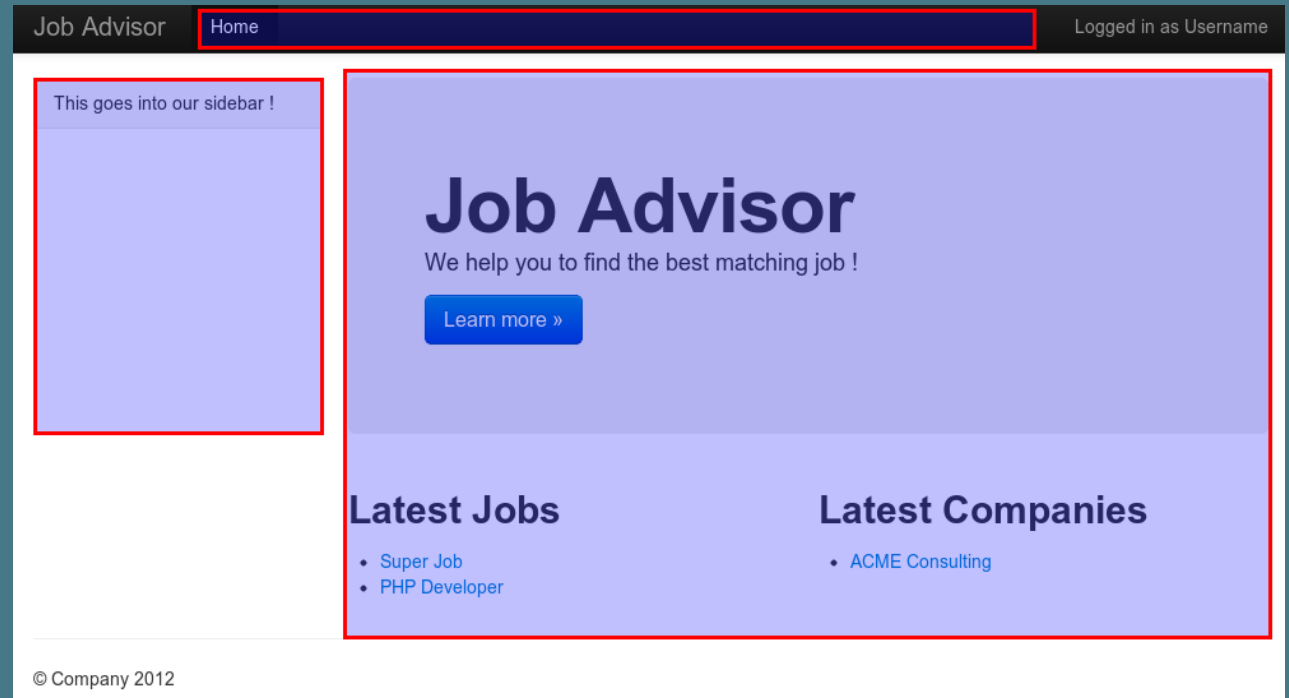
- tbd

- image(\$path, \$options)
- link(\$title, \$url, \$options)
- url(\$url, \$full) - wraps Router::url()
-
- ADD table examples



CakePHP Views: Creating a basic layout

- Using a basic Twitter Bootstrap template
 - Adding css resources
 - Adding title
 - Adding blocks
 - sidebar
 - Adding elements
 - navigation
 - Content



CakePHP Views: Elements & Blocks

- Using elements
 - Passing variables
 -
- Using blocks
 - start()
 - append()
 - prepend()
 - end()
 -

```
<a class="brand" href="#">Job Advisor</a>
<div class="nav-collapse collapse">
  <p class="navbar-text pull-right">
    Logged in as <a href="#" class="navbar-link">Username</a>
  </p>
  <?php echo $this->element('navigation'); ?>
</div><!--/.nav-collapse -->
```

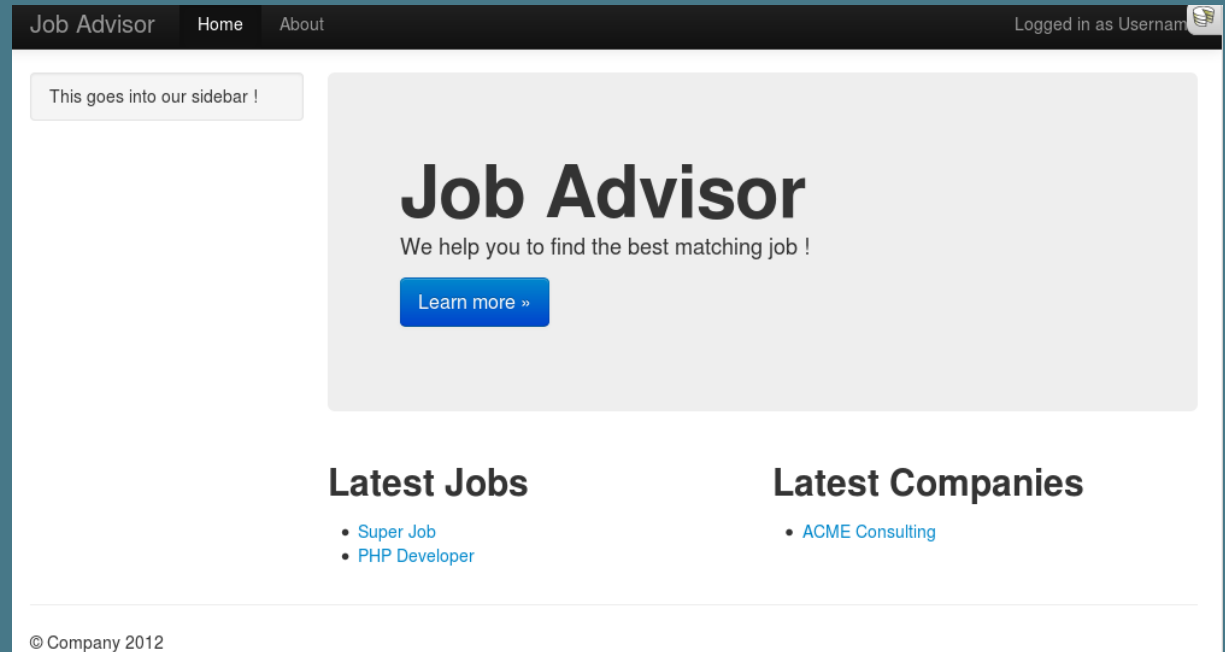
```
<div class="span3">
  <div class="well sidebar-nav">
    <?php echo $this->fetch('sidebar'); ?>
  </div><!--/.well -->
</div><!--/span-->
<div class="span9">
  <?php echo $this->fetch('content'); ?>
</div><!--/span-->
```

```
<?php $this->start('sidebar'); ?>
<ul class="nav nav-list">
  <li>This goes into our sidebar !</li>
</ul>
<?php $this->end(); ?>
```



CakePHP Finishing our Sample App

- Adding detail pages for
 - Jobs
 - Companies
 -



CakePHP Model Unit testing introduction

- CakePHP provides:
 - Integration with testing framework
 - You need to install phpunit first
 - Console shell to build test files
 - Console shell to build fixtures
 - Test datasource
- Bake Model tests for Company
 - Fixtures
 - setup() & tearDown()
 - TestMethods
 -
- Bake Fixtures
-
- Running our tests (Console & web)



CakePHP Model Unit testing – adding our first test

- tbd

