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

CakePHP

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
```

{tag: 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
```

{tag: bake-project}

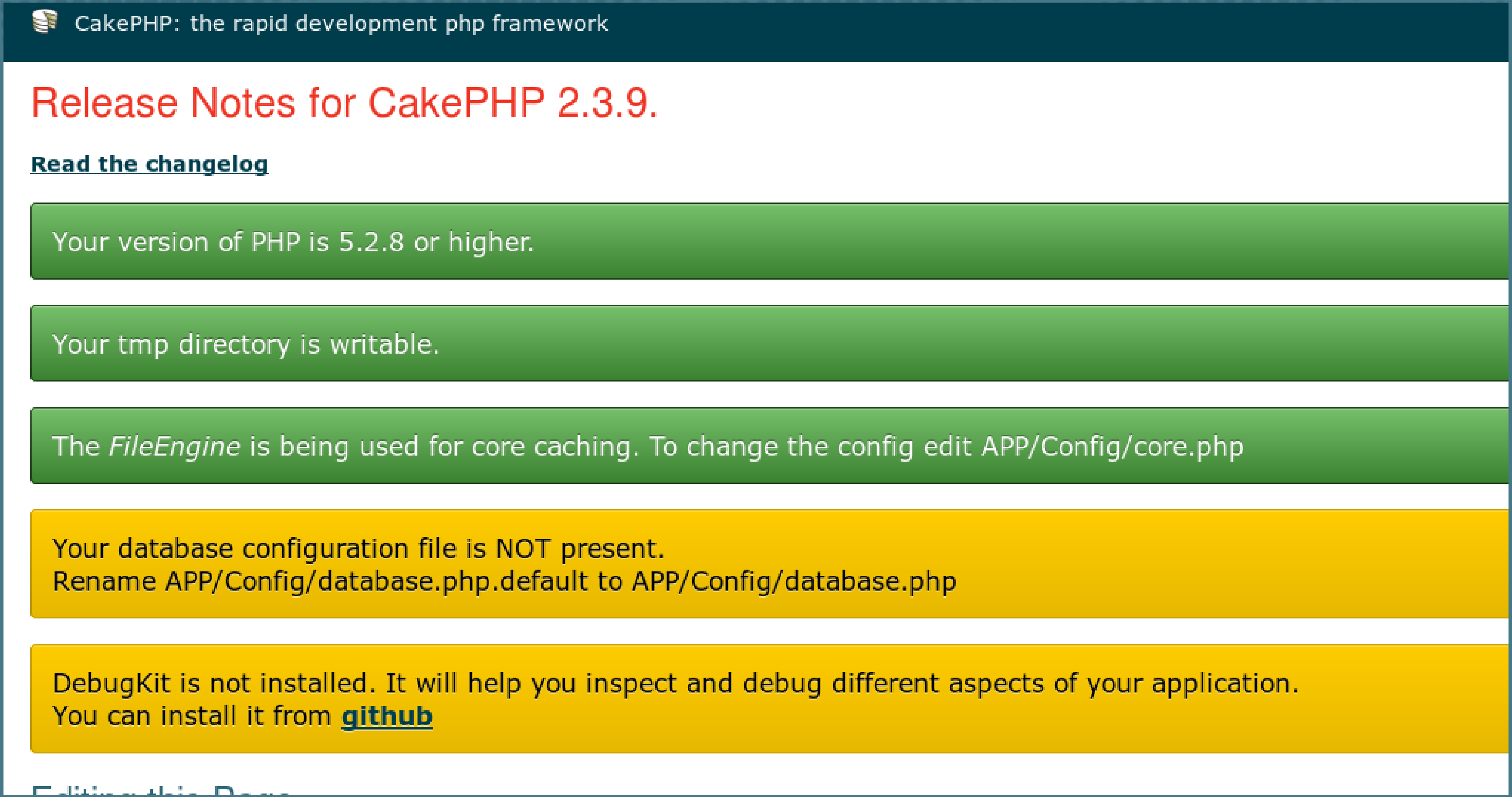
Development environment: Folder Structure

- 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

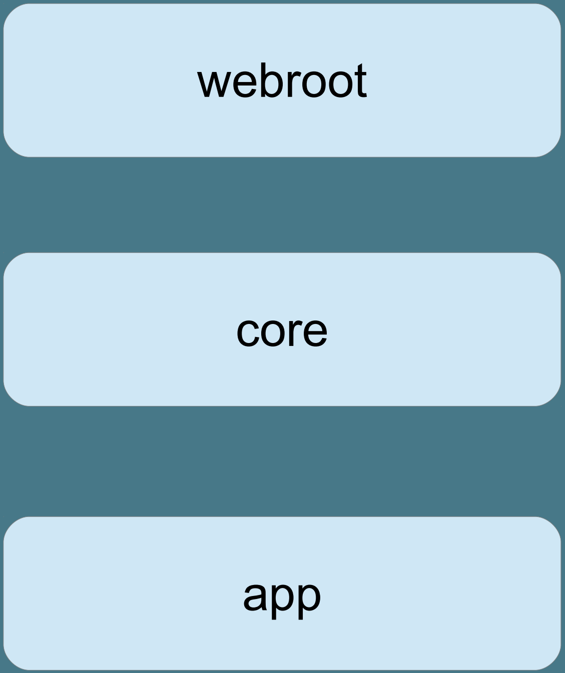
Your database configuration file is NOT present.
Rename APP/Config/database.php.default to APP/Config/database.php

DebugKit is not installed. It will help you inspect and debug different aspects of your application.
You can install it from **github**

Editing this Page

Non standard setups

Fix paths using webroot/index.php relevant constants



< Point here your webserver document root

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

< 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
```

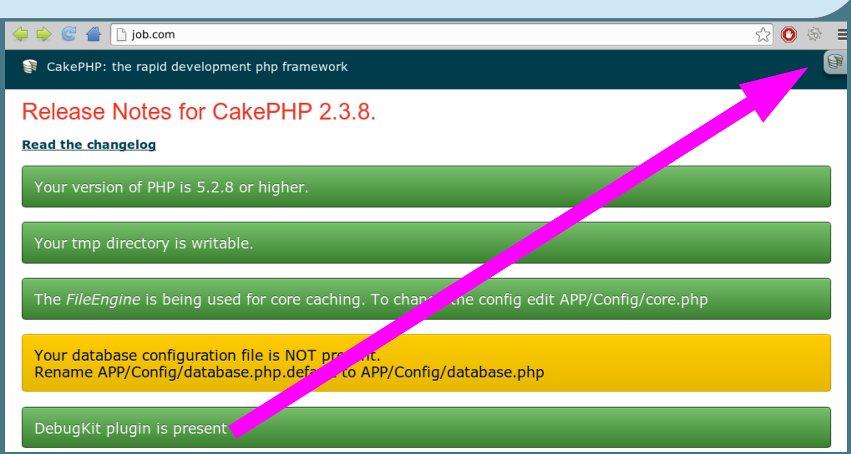
Configure your database.php file

{tag: debugkit}

Setting up DebugKit Plugin

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

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



CakePHP: the rapid development php framework

Release Notes for CakePHP 2.3.8.

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 chan the config edit APP/Config/core.php

Your database configuration file is NOT pr nt.
Rename APP/Config/database.php.def to APP/Config/database.php

DebugKit plugin is present

- Download / Submodule
- Load in Bootstrap

  CakePlugin::load('DebugKit');

- Load Component in AppController

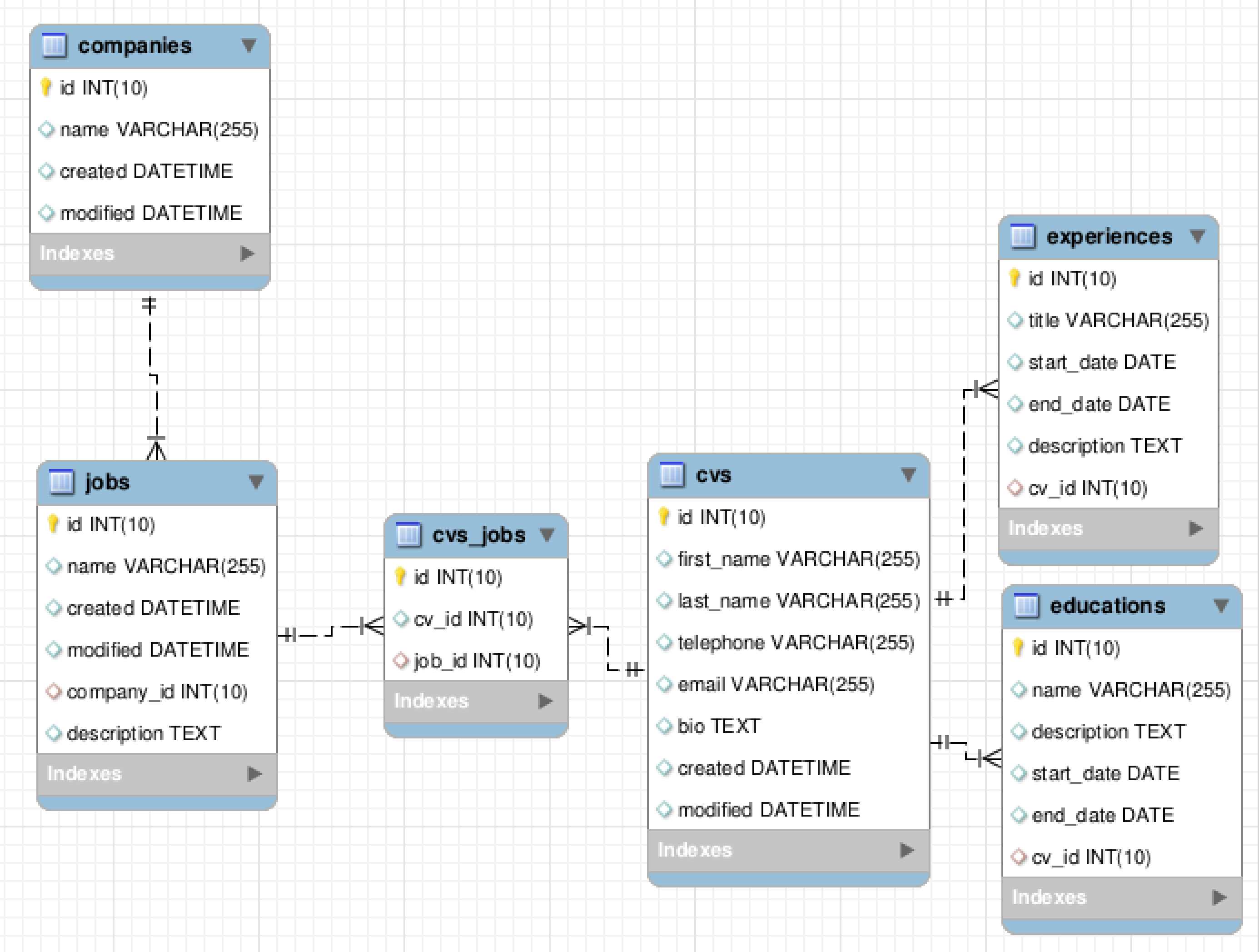
  public $components = array('Session', 'DebugKit.Toolbar');

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)



companies
- id INT(10)
- name VARCHAR(255)
- created DATETIME
- modified DATETIME

jobs
- id INT(10)
- name VARCHAR(255)
- created DATETIME
- modified DATETIME
- company_id INT(10)
- description TEXT

cvs_jobs
- id INT(10)
- cv_id INT(10)
- job_id INT(10)

cvs
- id INT(10)
- first_name VARCHAR(255)
- last_name VARCHAR(255)
- telephone VARCHAR(255)
- email VARCHAR(255)
- bio TEXT
- created DATETIME
- modified DATETIME

experiences
- id INT(10)
- title VARCHAR(255)
- start_date DATE
- end_date DATE
- description TEXT
- cv_id INT(10)

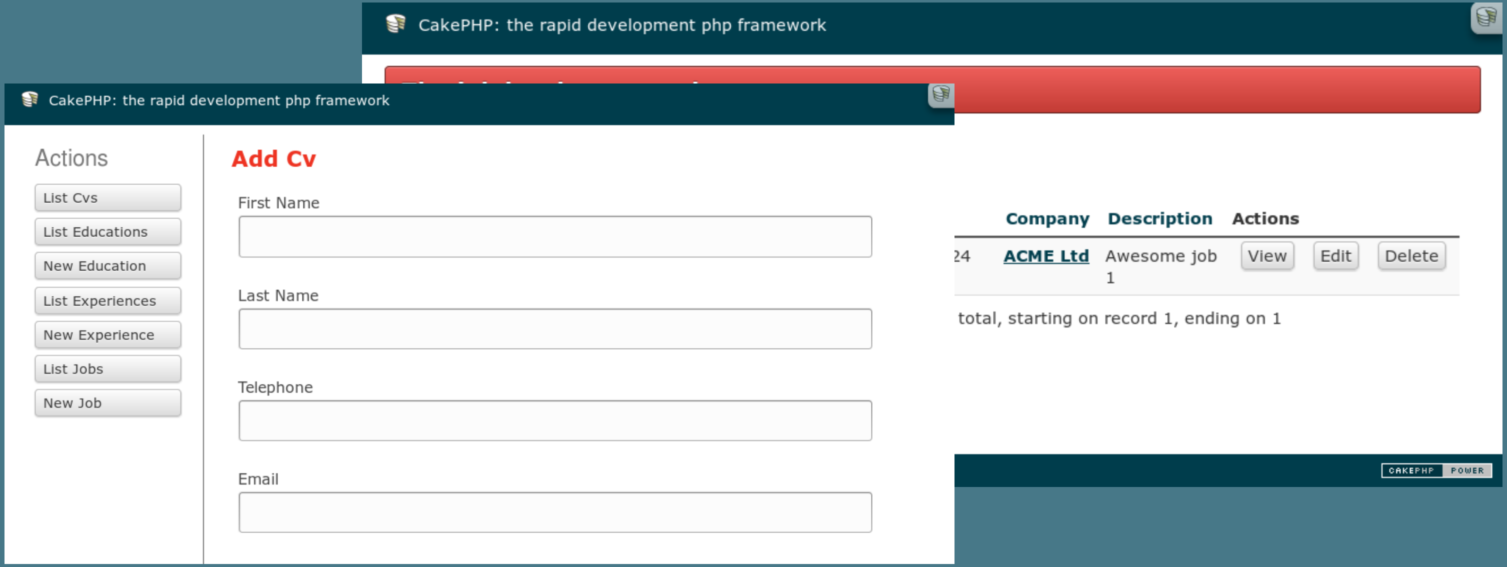
educations
- id INT(10)
- name VARCHAR(255)
- description TEXT
- start_date DATE
- end_date DATE
- cv_id INT(10)

{tag: cruds}

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



Add Cv

First Name

Last Name

Telephone

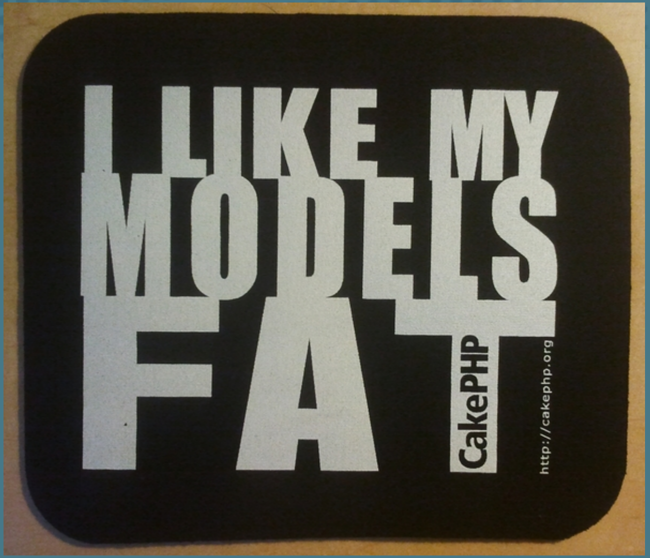
Email

```
Company    Description    Actions
24   ACME Ltd   Awesome job 1
total, starting on record 1, ending on 1
```

CakePHP Models: Introduction

- Business logic
- Validation
- Associations
- Usually linked to a database table

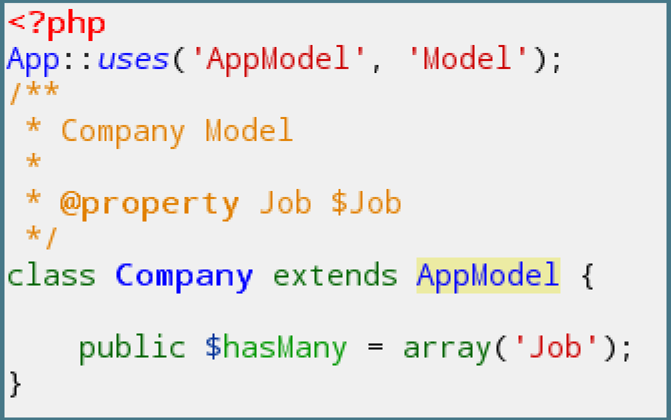
- Examples
  - Job (jobs)
  - CV (cvs)
  - Company (companies) ← note inflection



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');
}
```

{tag: display-name}

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

{tag: after-save}

CakePHP Models: Callbacks - email me on new Companies

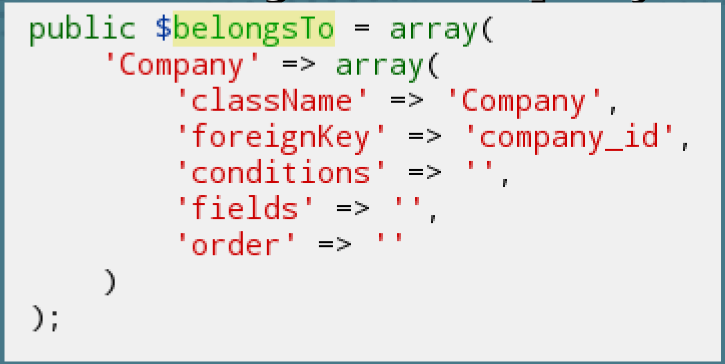
- Configure email.php
- afterSave($created)

```
public function afterSave($created) {
        if ($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

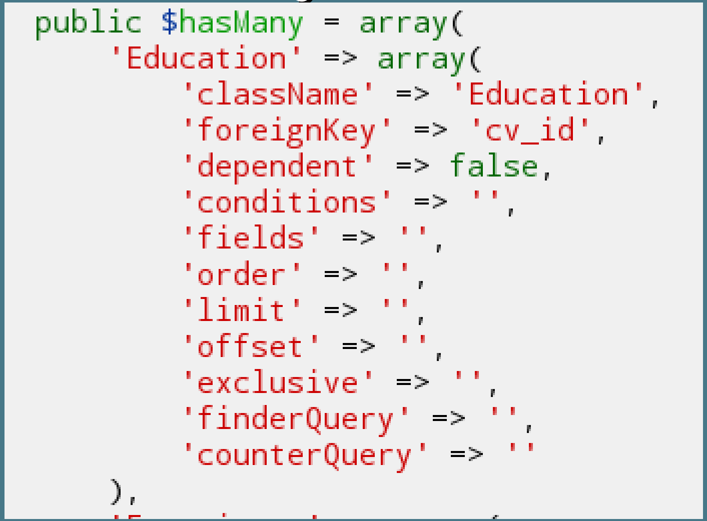
CakePHP Models: Associations - Intro and properties

- Job belongsTo Company



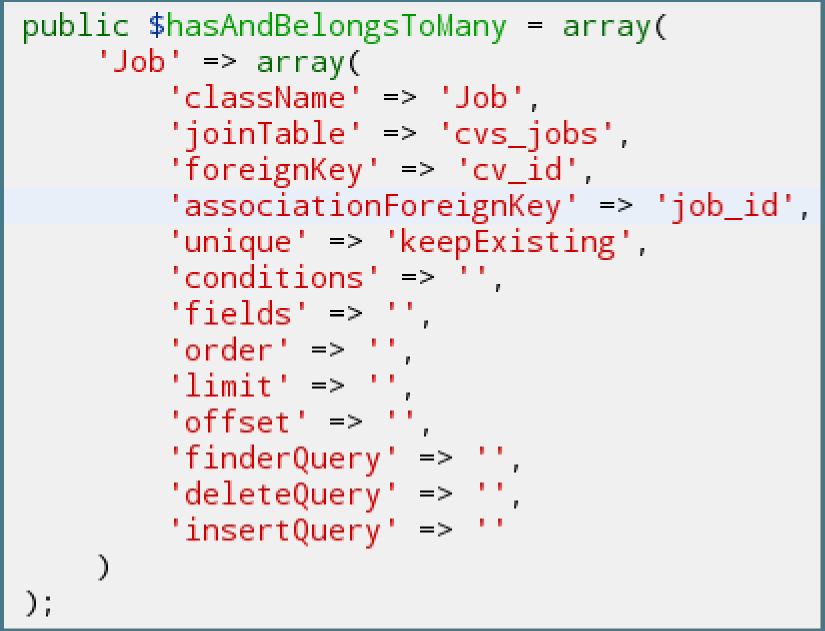
```
public $belongsTo = array(
    'Company' => array(
        'className' => 'Company',
        'foreignKey' => 'company_id',
        'conditions' => '',
        'fields' => '',
        'order' => ''
    )
);
```

- 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' => ''
    )
);
```

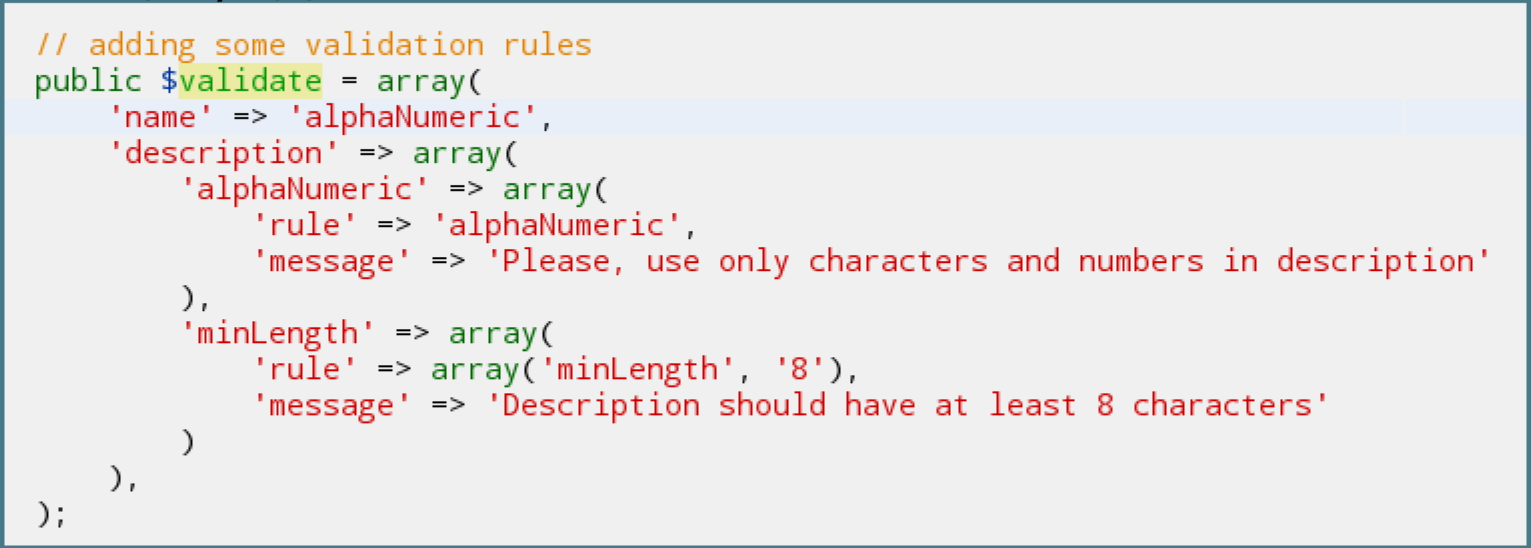
- Add an 'UnfinishedPositions' association to the Cv model

{tag: validation}

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'
        )
    ),
);
```

{tag: custom-validation}

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

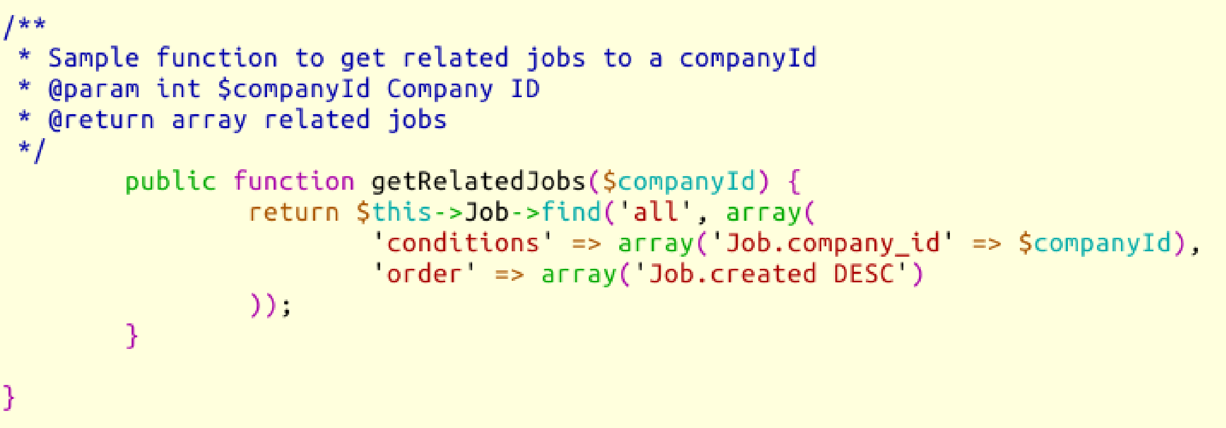
```
$jobs = $this->find('all');

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

```
conditions
recursive
fields
order
group
limit, page, offset
callbacks
```

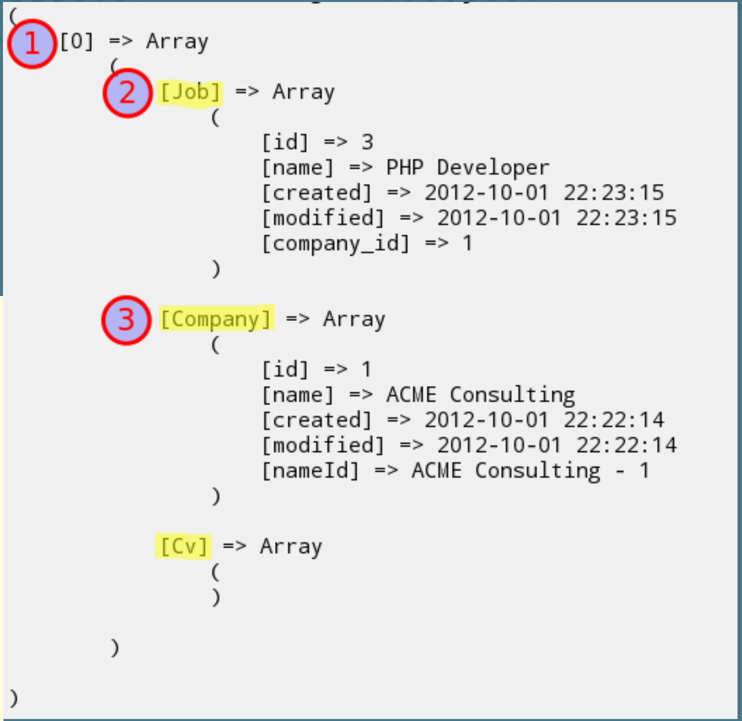
{tag: related-jobs}

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')
                ));
        }

}
```



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

            [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
                (
                )

        )

)
```

{tag: custom-finder}

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;
}
```

{tag: on-the-fly}

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()'),
            )
        )
));
```

{tag: on-the-fly}

Basic Workshop Session 2

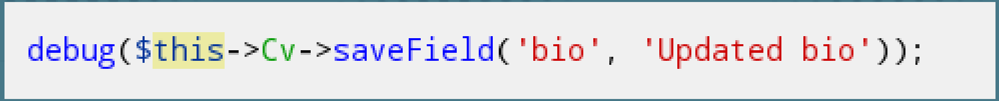
- What was covered yesterday
  - Setting up a development environment
  - Working with models
    - Validation
    - Callbacks
    - Associations
    - Reading data

- Slides are here >
- https://dl.dropboxusercontent.com/u/142951/basic-workshop.pdf
- Code repo is here > https://github.com/steinkel/cakefest-basic-workshop-2013
- Dump with some data is here >
- https://dl.dropboxusercontent.com/u/142951/dump_jobadvisor_data.sql

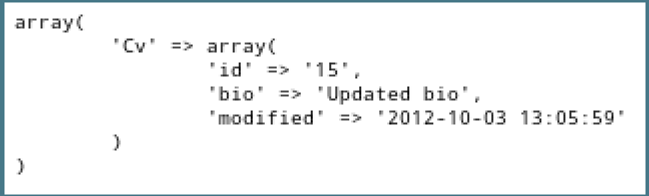
{tag: simple-save}

CakePHP Models: Saving data

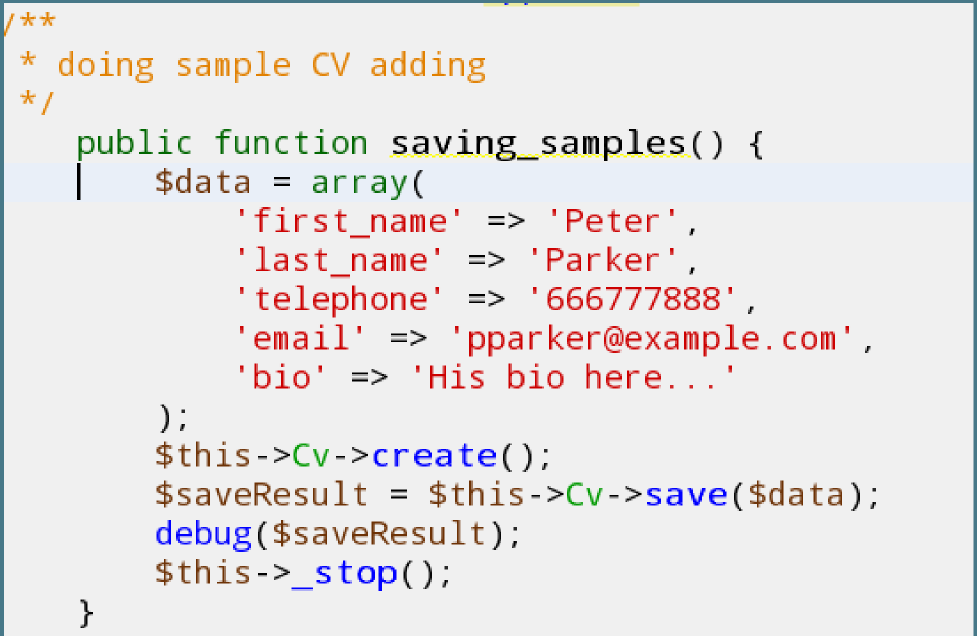
- create()
- save($data, $options)
- saveField('field', $value)
- saveMany($data)
- saveAssociated($data)
- saveAll($data)



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



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

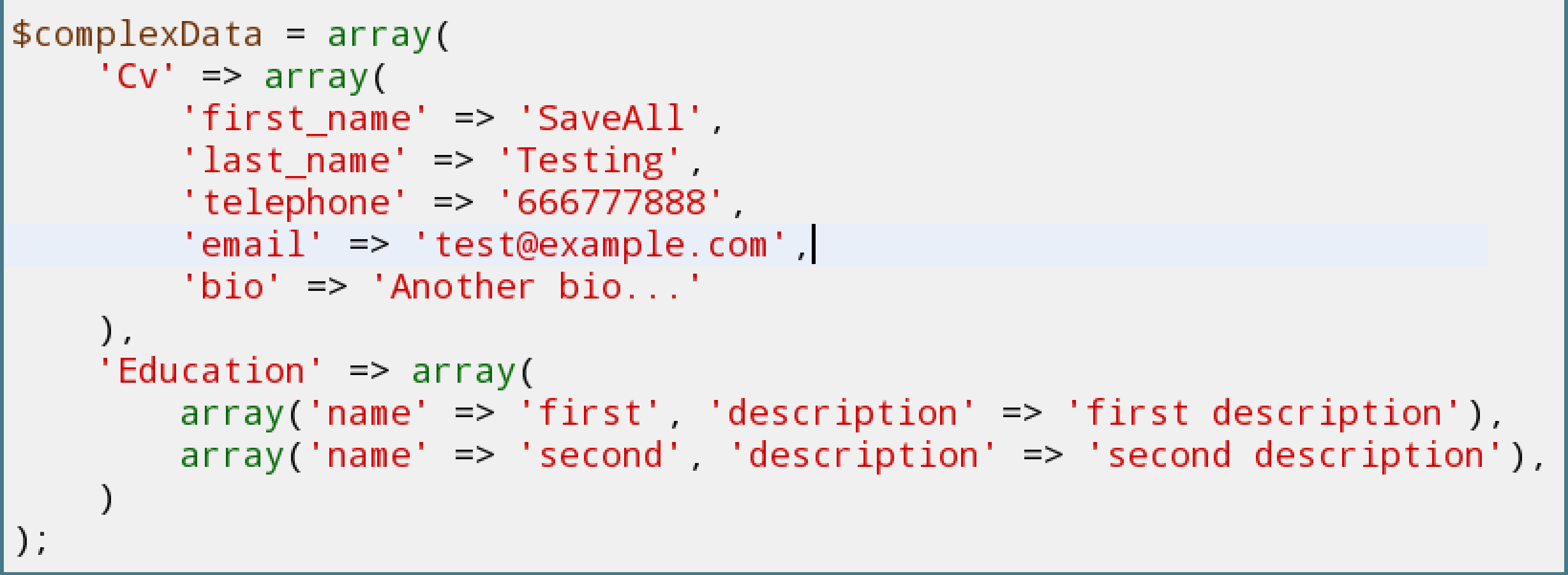


```php
/**
 * doing sample CV adding
 */
    public function saving_samples() {
        $data = array(
            'first_name' => 'Peter',
            'last_name' => 'Parker',
            'telephone' => '666777888',
            'email' => 'pparker@example.com',
            'bio' => 'His bio here...'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

{tag: complex-save}

CakePHP Models: Saving data – complex structures

- saveAssociated($data)
- 'deep' => true when needed



```
$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'),
    )
);
```

{tag: behavior}

CakePHP Behaviors: 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');
        }
    }

}
```

{tag: containable}

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'),
                    ))
            ))
```

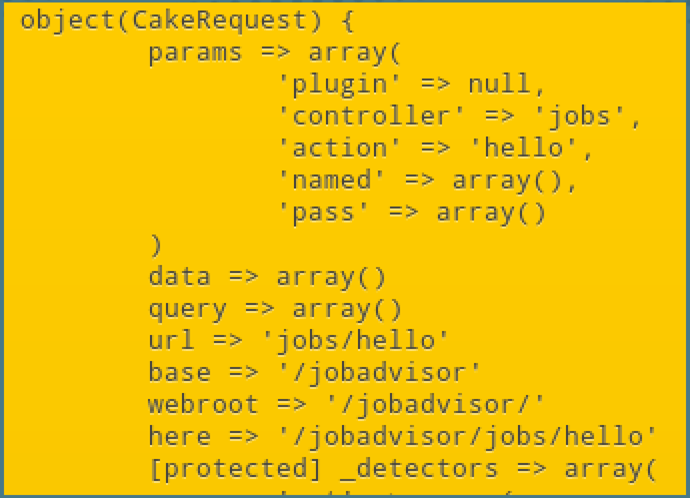
CakePHP Controllers: Introduction

- 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(
```

{tag: download}

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();

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

- Downloading the error log

CakePHP Views: Introduction

- Layouts
  - {Blocks/Elements}
  - Views
    - {Blocks/Elements}

- CakePHP default layout
- Static pages

- Helpers
  - Using helpers in views

CakePHP Views: Html Helper

- Available by default in all views
- Generates well formatted html elements, 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()
```

{tag: details}

CakePHP Views: Use Html Helper Example

- Create a new details action and view
- Action exactly the same as “view”
- Display when the company was created
- Display a list of all the associated Jobs and a link to their “details” action

```
foreach ($company['Job'] as $job) {
        $jobLinks[] = $this->Html->link($job['name'], array(
                'controller' => 'jobs',
                'action' => 'details',
                $job['id']
        ));
}
echo $this->Html->nestedList($jobLinks);
```

{tag: multiple-checkboxes}

CakePHP Views: Form Helper

- Creating forms
  - create($model, $options)
    - Detects $this->request->data model form id == null
      - <form id="ModelAddForm" method="post" action="/model/add">
    - Detects model form id != null
      - <form id="ModelEditForm" method="post" action="/models/edit/1">
      - <input type="hidden" name="_method" value="PUT" />

input($field, $options)

- text
- textarea
- checkbox
- password
- day, month, year, hour, minute, meridian

CakePHP Views: Other Core Helpers

- NumberHelper
  - format / currency
- SessionHelper
  - Flash messages
- TextHelper
  - excerpt / truncate
  - autolink
- TimeHelper
  - format time
  - deal with timezones
  - time checks

{tag: aliasing}

CakePHP Helpers: Aliasing Helpers

- Extending FormHelper
- Override the select method to force 'multiple' => 'checkbox'
- Override $this->Form in views, app wide

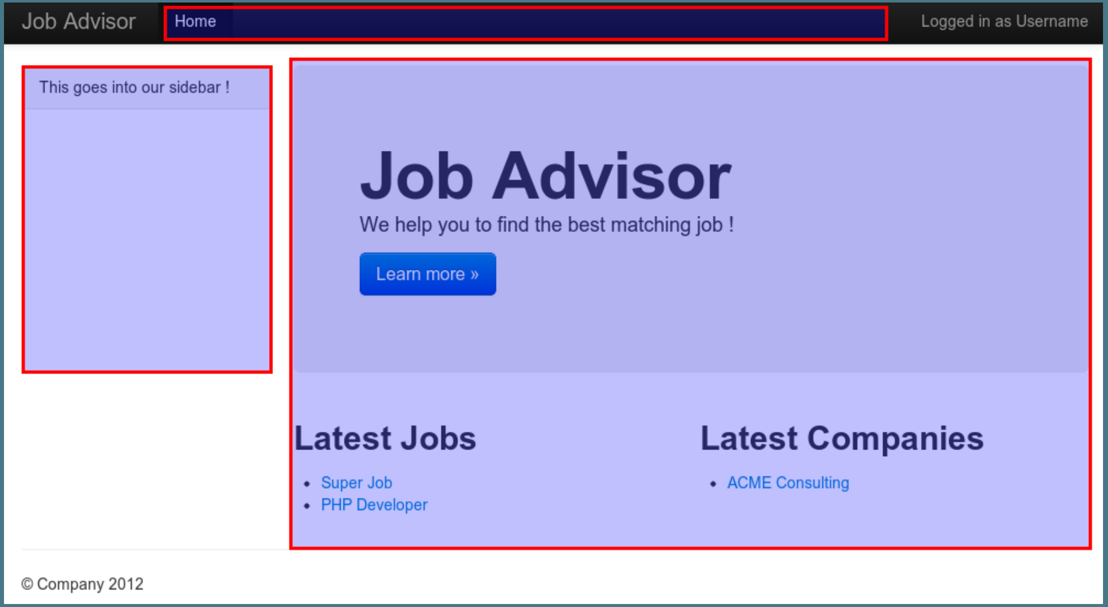
```
public function select($fieldName, $options = array(), $attributes = array()) {
    if (!empty($attributes['multiple'])) {
        $attributes['multiple'] = 'checkbox';
    }
    return parent::select($fieldName, $options, $attributes);
}
```

{tag: front-layout}

CakePHP Views: Creating a basic layout

- Using a basic Twitter Bootstrap template
- Copy/paste from >
  - http://getbootstrap.com/2.3.2/examples/hero.html

- Modify new “front” template
  - Adding css resources
  - Adding title
  - Adding blocks
    - sidebar
  - Adding elements
    - navigation
  - Content



Job Advisor

We help you to find the best matching job !

Latest Jobs

- Super Job
- PHP Developer

Latest Companies

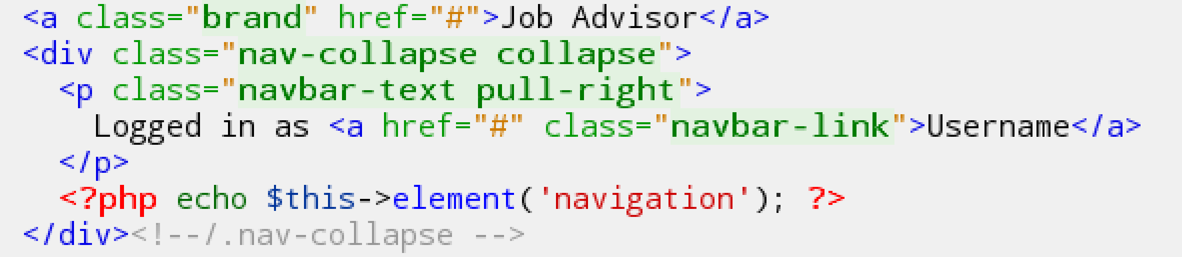
- ACME Consulting

© Company 2012

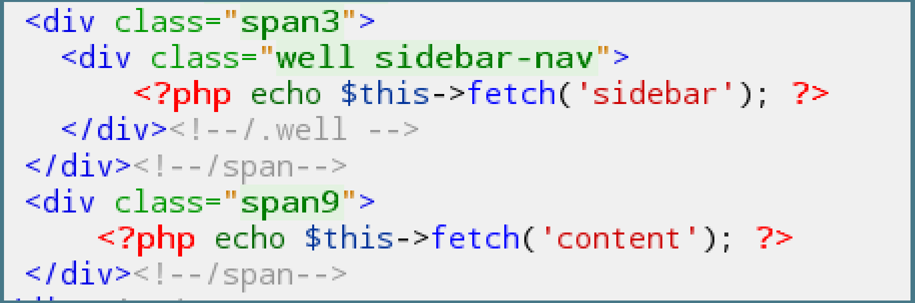
{tag: elements-home}

CakePHP Views: Elements & Blocks

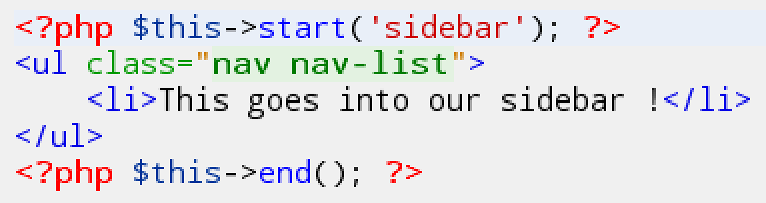
- Using elements
  - Passing variables
- Using blocks
  - start()
  - append()
  - prepend()
  - end()
- Adding a
  - jobs/home



```
<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 Model unit testing introduction

- CakePHP provides:
  - Integration with testing framework
    - You need to install phpunit first
  - Console shell to generate
    - test files
    - fixtures
  - Test datasource

  - Bake Fixtures… what?
  - Bake Model tests for Company
    - Fixtures… what?
    - setup() & tearDown()
    - TestMethods

  - Running our tests (Console & web)

CakePHP Model Unit testing – testing model method

- bake test case for Company Model (empty)
  - Console/cake bake test

- run test using the console
  - Console/cake test app Model/Company
- run test using the web runner
  - http://jobadvisor.com/test.php

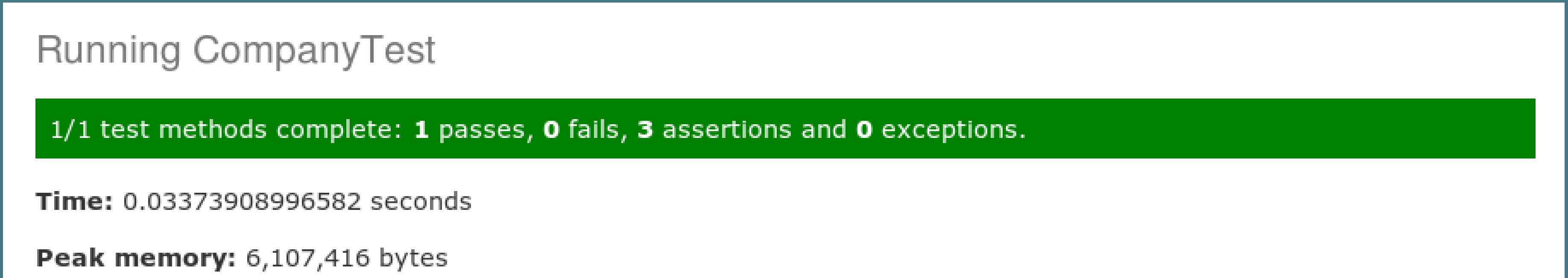
- think

- fix the fixtures
- implement the test case

{tag: fixtures}

CakePHP Model Unit testing – testing model method

- Add fix the fixtures and add some data



Running CompanyTest

1/1 test methods complete: **1** passes, **0** fails, **3** assertions and **0** exceptions.

Time: 0.033739089965820 seconds

Peak memory: 6,107,416 bytes

{tag: mock-mock}

CakePHP Model Unit testing – Mocking basics

- Create Mocked methods to test a Model method

```
public function testTodayIsWorkdaySuccessAndFail() {
		$jobMock = $this->getMockForModel('Job', array('getNow'));
		$jobMock->expects($this->at(0))
			->method('getNow')
			->will($this->returnValue(strtotime('next sunday')));
		$jobMock->expects($this->at(1))
			->method('getNow')
			->will($this->returnValue(strtotime('next thursday')));
		$result = $jobMock->todayIsWorkday(array());
		$this->assertFalse($result, 'next sunday should NOT be a workday');
		$result = $jobMock->todayIsWorkday(array());
		$this->assertTrue($result, 'next thursday should NOT be a workday');
	}
```

{tag: misc}

CakePHP Basic Workshop - Pick one...

- Security component essentials
- Using cake schema
- Adding a default / route to our application
- Cache intro
- Extending views

CakePHP Basic Workshop - Questions?

- Questions are welcome

Thank you!