Views

Jill Gundersen

Views

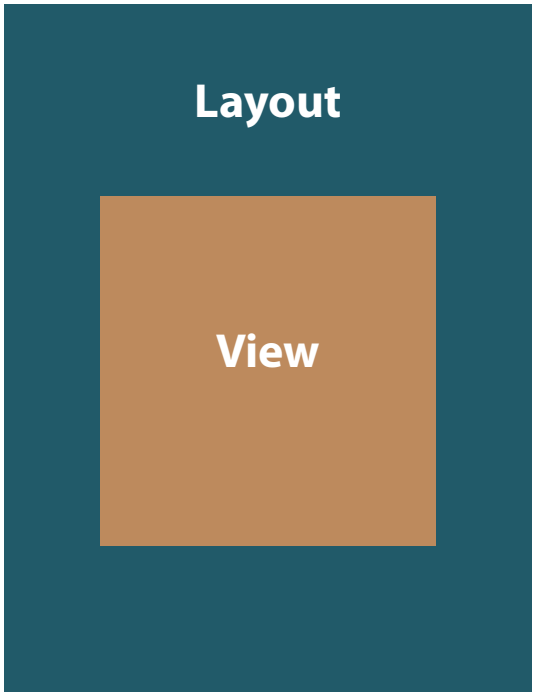
- **Presentation Layer**
  - Display the desired output from the request to the user.
- **Usually in HTML**
  - Can also be…
    - XML
    - JSON
    - RSS
    - Files and Streaming Files
- **CakePHP Template Files**
  - Written in plain PHP
  - .ctp ending
    - CakePHP supports other templating languages such as Smarty or Twig

Parts of the View Layer

- **View Layer Consists of 4 Parts**
  - layouts
    - The main layout for the website
  - views
    - Displays a unique response based on the action being run
  - elements
    - Reusable code rendered inside of a view
  - helpers
    - Provides view logic and helps build code for forms, pagination, etc.

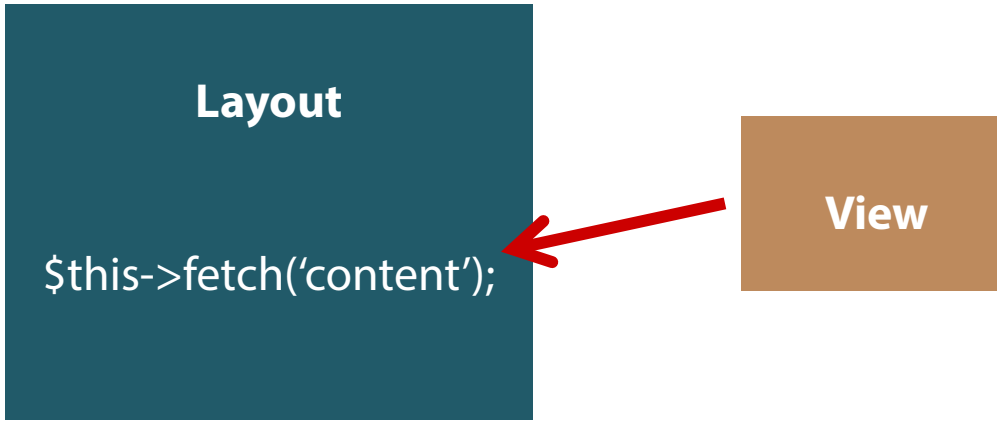
Layout

- **Presentation Layer**
  - Does not include the views
- **Contents**
  - Layout file contains the actual <html>,<head>, and <body> tags.
- **Wraps around your views**
- **Location**
  - /app/View/Layouts/
- **Default Layout**
  - default.ctp
  - /app/View/Layouts/default.ctp
- **Creating New Layout**
  - home.ctp
  - /app/View/Layouts/home.ctp



Rendered View

- **'content' Block**
  - Example
    - $this->fetch('content');
  - Contains the rendered views
    - index.ctp
    - view.ctp
- **Location?**
  - This 'content' block can be placed anywhere in the Layout
    - Typically in the main section of your layout



Page Title

- **Individual Page Title**
  - The layout <title> tag can be set on any page.
  - Using a specific a variable in the action of a controller or on the view template itself
- **Variable Name**
  - title_for_layout
  - Example
    - $this->set('title_for_layout', 'Our List of Delicious Cakes');

CSS and Images

- **CSS and Images**
  - Styles and images can be added to the site
  - Located:
    - /app/webroot/css – CSS Folder
    - /app/webroot/img – Image Folder
- **Linking CSS**
  - Example
    - $this->Html->css('nameOfStyleSheet');
    - No need to add the .css
  - Called in the layout template or the view template.
    - The view template call has a couple more parameters and we will cover that later in this module.
- **Displaying images**
  - Example
    - $this->Html->image('nameOfImage.jpg')
  - Used in any view template or layout
  - This will be covered in more depth later in this module

Multiple Layouts

- **Different Layouts**
  - Sometimes you need different layouts for different occasions
    - Sign up form, promotional, blog template
- **Layout Choice**
  - Can be set in the controller or the view template file
  - Example
    - $this->layout = ‘promotional’;
    - $this->layout = ‘default’;

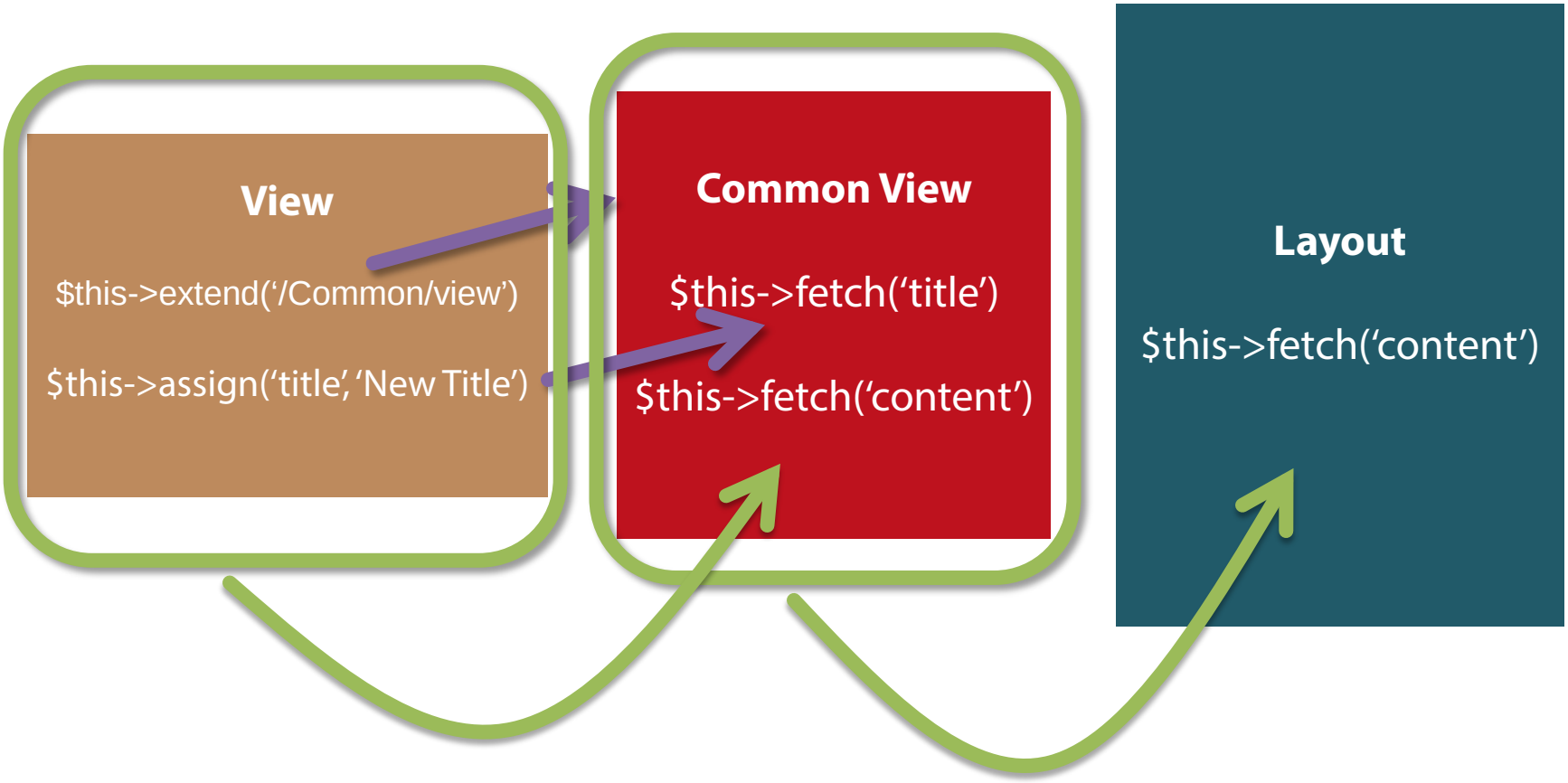
Views

- **Display Specific Content**
  - Specific content for a specific page
  - We created a few in our catalog site
  - Location
    - /app/Views/ControllerName/view_name.ctp
  - View names are based on the controller action
  - Override the default view by using
    - $this->render('view_name');

Extending Views

- **Extending Views**
  - Common views can be extended for use with detailed views
    - Detailed views are the regular views associated with the controller actions
  - Located in the app/Views directory inside the Common folder
    - app/Views/Common/view.ctp
  - Example of Common View
    - Same header layout for all of the page views, the only difference is the title
    - Example
      - $this->fetch('content') – mandatory for all common views
      - $this->fetch('title');
  - Detailed View
    - Extend the common view
    - Assign the title
    - Example
      - $this->extend('/Common/view');
      - $this->assign('title', 'New Title Name');

Extending Views – Visual Representation



Extending CSS

- **Specific Page/View CSS**
  - The layout contains a fetch call for CSS.
    - $this->fetch('css');
  - Normally in a view template you would "assign" the css
  - Different Call
    - This changed in version 2.4
    - $this->Html->css('cssFile', null, array('inline' => false));

Elements

- **Think Reusability**
  - Bits of Code
    - Available on different pages
    - Different locations
  - Examples:
    - Sub Navigation, Quote Box, Ads
- **Location**
  - app/View/Elements/
  - Template file (.ctp - just like all other view templates)

Using Elements

- **In Your View**
  - Call the element method and pass as the parameter your element view name
  - Example
    - echo $this->element('quote_box');
- **Passing Variables into an Element**
  - A second parameter can be passed into the element method
  - Associated array
    - key = the name of the variable
    - value = variable content
  - Example
    - echo $this->element('quote_box', array('quote' => 'CakePHP is awesome!');
    - echo $quote;

Helpers

- **What Are Helpers?**
  - "Helpers are component-like classes for the presentation layer of [our] application" – CakePHP Site
  - They help create "well-formed markup"
  - Basically they produce nuggets of code that we constantly use.
- **Include in Controller**
  - All helpers need to be included in the Controller.
  - Set the variable $helpers in either the current controller or the AppController
  - Example
    - public $helpers = array('Form', 'Html')

Helpers

- **Use**
  - We have used some of these Helpers throughout our Catalog site.
  - CakePHP has a number of Helpers, here is a list of them with the parameter value:
    - CacheHelper (Cache)
    - FormHelper (Form)
    - HtmlHelper (Html)
    - JsHelper (Js)
    - NumberHelper (Number)
    - Paginator – special scenario
    - RSS (Rss)
    - SessionHelper (Session)
    - TextHelper (Text)
    - TimeHelper (Time)

Helpers

- **Use**
  - We have used some of these Helpers throughout our Catalog site.
  - CakePHP has a number of Helpers, here is a list of them with the parameter value:
    - CacheHelper (Cache)
    - **FormHelper (Form)**
    - **HtmlHelper (Html)**
    - JsHelper (Js)
    - NumberHelper (Number)
    - Paginator – special scenario
    - RSS (Rss)
    - SessionHelper (Session)
    - TextHelper (Text)
    - TimeHelper (Time)

HtmlHelper

- **Basic Function of the HtmlHelper**
  - The HtmlHelper generates code that is “well-formated” and used often.
  - Helps with tags we often forget the syntax to
    - For Example: Style sheets - We know it’s a link tag, but does the link tag take a “href” for the source file or a “src”?
- **Lots of Methods**
  - The HtmlHelper has a lot of helpful methods, but we are only going to cover the following:
    - css
    - image
    - script
  - We will cover the basics of each
- **Link to HtmlHelpers**
  - http://book.cakephp.org/2.0/en/core-libraries/helpers/html.html

HtmlHelper - CSS

- **Updated**
  - Method has changed in version 2.4
    - This will not work with the current version of CakePHP we are working with.
    - If you have version 2.3 installed please refer to the segment on Views earlier in this module.
  - Information below is for the 2.4 version.
- **Takes One or Two Parameters**
  - Focus only on the first parameter, second is optional
- **First Paramenter**
  - CSS file name or an array of CSS file names
    - This is relative to the app/webroot/css folder
  - Example
    - echo $this->Html->css('styleSheet'); // .css file extention not needed
    - echo $this->Html->css(array('menus', 'layout')); // .css file extention not needed
  - Output
    - <link rel="stylesheet" type="text/css" href="/css/menus.css" />
    - <link rel="stylesheet" type="text/css" href="/css/layout.css" />

HtmlHelper - Image

- **Parameters**
  - First – string path to the image
    - This is relative to the app/webroot/img folder
  - Second – optional associated array of options (html attributes)
- **Example**
  - echo $this->Html->image('chocCake.jpg', array('alt' => 'Chocolate Cake');
- **Output**
  -

HtmlHelper - Script

- **Similar to the CSS Method**
- **Parameters**
  - First – String to a single JS file, or an array of JS files.
    - This is relative to the app/webroot/js folder
    - Will also allow for directories outside webroot/js folder
    - Can also take a path to a remote URL
  - Second – optional associated array of options (html attributes)
  - We will only focus on the first parameter
- **Example**
  - echo $this->Html->script("scripts");
  - echo $this->Html->script("/newJSDirectory/scripts");
  - echo $this->Html->script("http://www.somesites.com/jsFile.js");
    - Notice you must include the extention
- **Output**
  - <script type="text/javascript" src="/js/scripts.js"></script>

FormHelper

- **Basic Function of the FormHelper**
  - The FormHelper generates the needed code for form creation
  - Creates forms quickly
  - Streamlines validation, re-population and layout.
- **Lots of Methods**
  - The FormHelper has a lot of methods. We will be covering the following:
    - create
    - end
    - hidden
    - password
    - input
  - We will cover the general basics of each of them
- **Link to FormHelper**
  - http://book.cakephp.org/2.0/en/core-libraries/helpers/form.html

FormHelper - Create

- **Parameters**
  - First – optional string model name
  - Second – optional associated array of options
- **Defaults**
  - Form method defaults to ‘post’
- **Example**
  - General Add or Edit form with Model
    - echo $this->Form->create(“Items”);
- **Output**
  - <form id=“ItemAddForm” method=“post” action=“/items/add”>
  - If Edit form
    - <form id=“ItemEditForm” method=“post” action=“/items/edit/5”>
    - <input type=“hidden” name=“_method” value=“PUT” />

FormHelper - End

- **Single Parameter Optional**
  - String name for submit button or
  - Associated array of options
- **Example**
  - echo $this->Form->end();
  - echo $this->Form->end("Add Item");
- **Output**
  - </form>
  -
    <input type="submit" value="Add Item" />

    </form>

FormHelper - Hidden

- **Parameters**
  - First - string field name
  - Second – optional associated array of options
- **Example**
  - echo $this->Form->hidden(“id”);
- **Output**
  - <input name=“data[Item][id]” value=“16” id=“ItemId” type=“hidden” />

FormHelper - Password

- **Parameters**
  - First - string field name
  - Second – optional associated array of options
- **Example**
  - echo $this->Form->password(“password”);
- **Output**
  - <input name=“data[User][password]” value=“” id=“UserPassword” type=“password” />

FormHelper - Input

- **Parameters**
  - First – string field name
  - Second – optional associated array of options
- **Basic Understanding**
  - Output of an input method
    - div container
    - label tag
    - input tag
    - error element (if applicable)

FormHelper - Input

- **Basic Understanding Cont'd**
  - The input method determines what type of input you need based on the field that is provided (first parameter).
  - Below is a list of associations based on column type

Column Type	Form Field
string (char, varchar, etc.)	text
boolean, tinyint(1)	checkbox
text	textarea
text (field name of password, passwd, psword)	password
text (field name of email)	email
text (field name of tel, telephone, phone)	tel
date	day, month, and year selects
datetime	day, month, year, hour, minute, and meridian selects
time	hour, minute, and meridian selects

FormHelper - Input

- **Input Options**
  - Optional
  - Associated array, key/value pairs
  - Below is a list of a few different types of options available. The full list can be found here
    - http://book.cakephp.org/2.0/en/core-libraries/helpers/form.html#options
- **Options List**
  - Type – Force the type of input to be used
    - Example - 'type' => 'email'
  - Label – provide personal text for the label tag
    - Example – 'label' => 'First Name'
  - Default – set the default value of the field
    - Example – 'default' => 'Chocolate Cake'
  - Selected – set a selected option based on the value provided
    - Example – 'selected' => '3'
  - Rows, Cols – set the rows and columns of a text area
    - Example – 'rows' => '5', 'cols' => '10'

FormHelper - Input

- **Example – Add an Item Form**

```php
<?php
echo $this->Form->create('Item');
echo $this->Form->input('category_id');
echo $this->Form->input('title');
echo $this->Form->input('year');
echo $this->Form->input('length');
echo $this->Form->input('description', array('rows' => '5'));
echo $this->Form->end('Add Item');
?>
```

- **Example – Edit an Item Form**

```php
<?php
echo $this->Form->create('Item');
echo $this->Form->input('category_id');
echo $this->Form->input('title');
echo $this->Form->input('year');
echo $this->Form->input('length');
echo $this->Form->input('description', array('rows' => '5'));
echo $this->Form->input('id', array('type' => 'hidden'));
echo $this->Form->end('Update Item');
?>
```

Summary

- **Four Parts of the View**
  - Layout
  - Views
  - Elements
  - Helpers
- **Layout**
- **Views**
  - Extending Views
- **Elements**
  - Reusable code
- **Helpers**
  - Code shortcuts
  - HtmlHelper, FormHelper