CodeIgniter PHP MVC Framwork

OSSF, 自由軟體鑄造場 ,Who's Who

吳柏毅　appleboy@OpenFoundry workshop

Taiwan Community

- **http://phorum.study-area.org/ 酷！學園**
- **http://www.openfoundry.org/ OSSF, 自由軟體鑄造場**
- **http://whoswho.openfoundry.org/forum.html Who's Who**

Who am I?

- I am appleboy (My nick name) 吳柏毅
- Google appleboy keyword
- Blog: http://blog.wu-boy.com
- 酷學園： appleboy
- Ptt BBS: appleboy46
- Twitter: appleboy
- Plurk: appleboy46

CodeIgniter Taiwan Community

- **Irc: Freenode #CodeIgniter.tw**
- **http://ci.wuboy.twbbs.org/** 台灣官方網站
- **http://ci.wuboy.twbbs.org/forum/**
  繁體中文討論區
- **http://ci.wuboy.twbbs.org/user_guide/**
  繁體中文文件

Popular PHP MVC Framework

- Zend Framework
- CakePHP
- Symfony (少數)
- CodeIgniter (少數)
- Kohana was forked from CodeIgniter 1.5.4 in 2007 (support php5)

PHP Web Framework Performance

- ZendFramework, CodeIgniter, CakePHP
- With APC PHP code cache
- Url : http://avnetlabs.com/php/php-framework-comparison-benchmarks

	Run 1	Run 2	Run 3	Run 4	Average
CakePHP	7.3	7.3	7.3	7.3	**7.3**
CodeIgniter	97.5	98.0	96.6	98.3	**97.6**
Zend Framework	32.8	33.3	31.8	32.7	**32.7**

Why Gallery3 use Kohana Framework

- Kohana was forked from CodeIgniter 1.5.4 in 2007. It base on PHP5.
- Zend Framework is 1705 files of framework code. It is 200-300% slower. ZF docs are comprehensive, but overwhelming and lacking in good examples making ZF development a very frustrating experience.
- CodeIgniter doesn't support exceptions, but Kohana has slightly worse performance than CI.

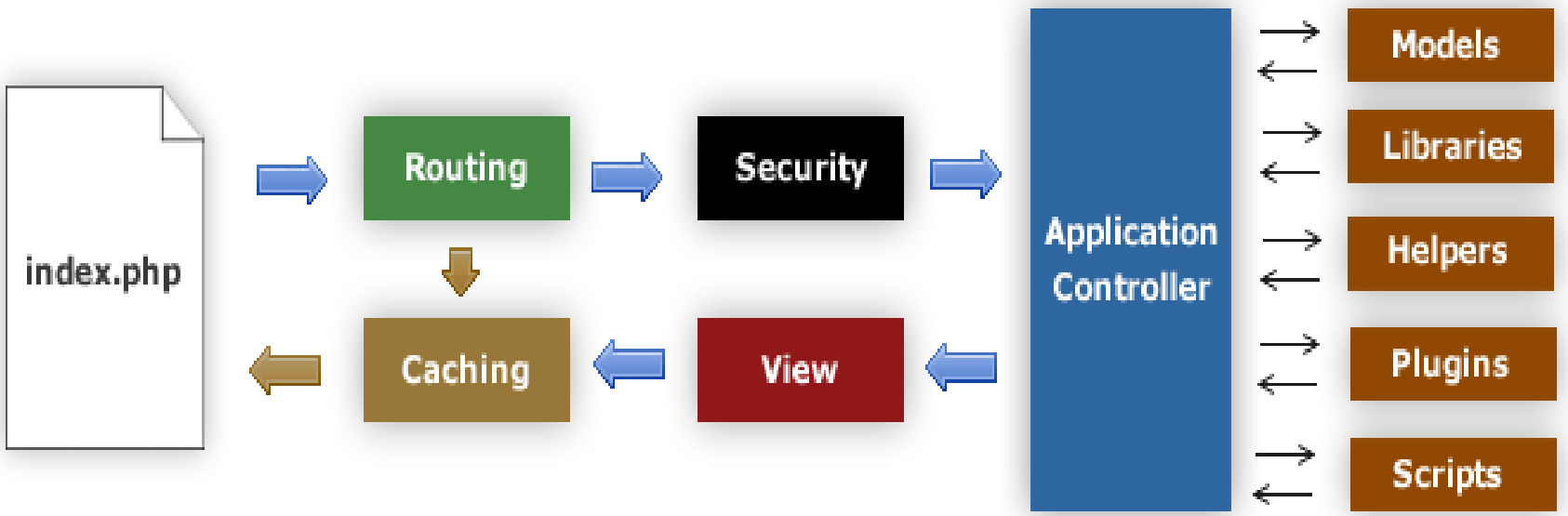
CodeIgniter is right for you if:

- You want a framework that does not require you to use the command line.
- You want a framework with a small footprint.
- You need exceptional performance.
- You need clear, thorough documentation.

CodeIgniter Features

- Model-View-Controller Based System
- PHP 4 Compatible
- Active Record Database Support
- Form and Data Validation
- Security and XSS Filtering
- Pagination
- Scaffolding
- Cache
- Large library of "helper" functions

Application Flow Chart



Model-View-Controller

- Model:
  - help you retrieve, insert, and update information in your database.
- View:
  - the information that is being presented to a user.
- Controller:
  - an intermediary between the Model, the View
  - any other resources needed to process the HTTP request and generate a web page

How to install it ? It is very easy

- Install Apache + PHP + MySQL (Appserv)
- Download CodeIgniter 1.7.1
- Unzip it, and move CI directory to www
- Open your browser http://127.0.0.1

You got it


Welcome to CodeIgniter!
The page you are looking at is being generated dynamically by CodeIgniter.
If you would like to edit this page you'll find it located at:
system/application/views/welcome_message.php
The corresponding controller for this page is found at:
system/application/controllers/welcome.php
If you are exploring CodeIgniter for the very first time, you should start by reading the User Guide.
Page rendered in 0.0395 seconds

The system/ Folder

- Application
- Cache
- Codeigniter
- Database
- Fonts
- Helpers
- Language
- Libraries
- Logs
- Plugins

The system/application Folder

- Config
- Controllers
- Errors
- Helpers
- Hooks
- Language
- Libraries
- Models
- views

How to create multiple project

- Edit index.php
  - $application_folder = "application";
  - $system_folder = "system";
- Move CI Core to www.

Exercise

- Create two project.
  - Foo
  - Bar
- Use common CI Core
- If you want upgrade CI, you can replace CI Core directory.

Initial Configuration: config.php

- system/application/config/
  - $config['base_url'] = 'http://localhost/';
  - $config['index_page'] = '';
    - To make this work, you need to include an .htaccess file to the CodeIgniter root directory.

default settings

```
$config['charset'] = “UTF-8”;
$config['cache_path'] = '';
$config['permitted_uri_chars'] = 'a-z 0-9~%.:_-';
$config['log_date_format'] = 'Y-m-d H:i:s';
$config['global_xss_filtering'] = TRUE;
```

CodeIgniter URLs

- example.com/index.php/news/article/my_article
  - news – Controller
  - article – class function
  - my_article - any additional segments
- If you do not use CodeIgniter, your urls is:
- example.com/news.php?mode=show&id=1

Removing the index.php file (Apache)

- By default, the index.php file will be included in your URLs:
- example.com/index.php/news/article/my_article
- using a .htaccess file with some simple rules
  - Edit httpd.conf
  - Unmark LoadModule rewrite_module modules/mod_rewrite.so
  - RewriteEngine on
  - RewriteCond $1 !^(index\.php|images|robots\.txt)
  - RewriteRule ^(.*)$ /index.php/$1 [L]

Removing the index.php file (Lighttpd)

- Edit lighttpd.conf

```
$HTTP["host"] == "mimi.twgg.org" {
  server.document-root = "/var/www/html/MLB/"
  url.rewrite = (
        "^/images/.*$" => "$0",
        "^/includes/.*$" => "$0",
        "^/(.*)$" => "index.php/$1"

  )
  accesslog.filename = "/var/log/lighttpd/mimi.twgg.org-access_log"
}
```

Adding a URL Suffix

- Edit config/config.php file
- example.com/index.php/products/view/shoes
- example.com/index.php/products/view/shoes.html

Enabling Query Strings

- In some cases you might prefer to use query strings URLs:
  - index.php?c=products&m=view&id=345
- index.php?c=controller&m=method
  - $config['enable_query_strings'] = FALSE;
  - $config['controller_trigger'] = 'c';
  - $config['function_trigger'] = 'm';
- Reduce the google search keywords

What is a Controller?

- example.com/index.php/blog/
- Then save the file to your application/controllers/ folder

```
<?php
class Blog extends Controller {

	function index()
	{
		echo 'Hello World!';
	}
}
?>
```

Controller

- Class names must start with an uppercase letter. In other words
- class Blog extends Controller { }

Functions

- example.com/index.php/blog/index/
- example.com/index.php/blog/comments/
-

```php
<?php
class Blog extends Controller {

	function index()
	{
		echo 'Hello World!';
	}

	function comments()
	{
		echo 'Look at this!';
	}
}
?>
```

Function Calls

- $this->$method();
-
```php
<?php
class Blog extends Controller {

	function index()
	{
		echo 'Hello World!';
	}

	function comments()
	{
		echo 'Look at this!';
	}
}
?>
```

Passing URI Segments to your Functions

- example.com/index.php/products/shoes/sandals/123
-

```php
<?php
class Products extends Controller {

    function shoes($sandals, $id)
    {
        echo $sandals;
        echo $id;
    }
}
?>
```

Defining a Default Controller

- open your application/config/routes.php file and set this variable:
- $route['default_controller'] = 'Blog';

URI Routing: routes.php

- Routing rules are defined in your application/config/routes.php file
- example.com/class/function/id/
- http://www.example.com/site/pages/4
- http://www.example.com/about_us/
- $route['about_us'] = “site/pages/4”;
- $route['blog/joe'] = "blogs/users/34";

Regular Expressions

- $route['products/([a-z]+)/(\d+)'] = "$1/id_$2";
- $route['default_controller'] = 'welcome';

Class Constructors

- In PHP4

```php
<?php
class Blog extends Controller {

        function Blog()
        {
            parent::Controller();
        }
}
?>
```

- In PHP5

```php
<?php
class Blog extends Controller {

        function __construct()
        {
            parent::Controller();
        }
}
?>
```

Exercise

- Remote index.php with url.
- Write controller News
- News contains showNewsList and showNews function.
- Use $this->$method(); to call showNewsList and showNews function in index function, then echo 『 hello world 』 。
- Set upload directory value in Class Constructors
  - $this->upload_folder = 'upload/news/';

Creating a View

- save the file in your application/views/ folder

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;
charset=utf-8" />
<title>My Blog</title>
</head>
<body>
    <h1>Welcome to my Blog!</h1>
</body>
</html>
```

Loading a View

- $this->load->view('name');
- example.com/index.php/blog/

```
<?php
class Blog extends Controller {

    function index()
    {
        $this->load->view('blogview');
    }
}
?>
```

Loading multiple views

```
<?php

class Blog extends Controller {

   function index()
   {
      $data['page_title'] = 'Your title';
      $this->load->view('header');
      $this->load->view('menu');
      $this->load->view('content', $data);
      $this->load->view('footer');
   }
}
?>
```

Storing Views within Sub-folders

- $this->load->view('folder_name/file_name');
- folder_name is Controller name

Adding Dynamic Data to the View

- You can use an array or an object in the second parameter of the view loading function.

```
$data = array(
            'title' => 'My Title',
            'heading' => 'My Heading',
            'message' => 'My Message'
        );

$this->load->view('blogview', $data);
```

```
$data = new Someclass();
$this->load->view('blogview', $data);
```

example

```
<?php
class Blog extends Controller {

    function index()
    {
        $data['title'] = "My Real Title";
        $data['heading'] = "My Real Heading";

        $this->load->view('blogview', $data);
    }
}
?>
```

```
<html>
<head>
<title><?php echo $title;?></title>
</head>
<body>
     <h1><?php echo $heading;?></h1>
</body>
</html>
```

Creating Loops: Controller

```
<?php
class Blog extends Controller {

    function index()
    {
        $data['todo_list'] = array('Clean House', 'Call Mom',
'Run Errands');

        $data['title'] = "My Real Title";
        $data['heading'] = "My Real Heading";

        $this->load->view('blogview', $data);
    }
}
?>
```

Creating Loops: view

```
<html>
<head>
<title><?php echo $title;?></title>
</head>
<body>
<h1><?php echo $heading;?></h1>
<h3>My Todo List</h3>
<ul>
<?php foreach($todo_list as $item):?>
<li><?php echo $item;?></li>
<?php endforeach;?>
</ul>
</body>
</html>
```

Returning views as data

- $string = $this->load->view('myfile', '', true);
- When to use it?

Exercise

- Create application/views/header.php and footer.php
- 傳入任意值 n, m 到 function number($num_1, $num_2)，輸出 n*1 ~n*m, output file application/views/output/index.php
- example.com/output/number/6/5
- example.com/output/number/10/4

Helper Functions

- $this->load->helper('name');
- to load the URL Helper file, which is named url_helper.php, you would do this:
- $this->load->helper('url');

Loading Multiple Helpers

- $this->load->helper(array('helper1', 'helper2', 'helper3'));

How to Auto-loading Helpers?

- To autoload resources, open the application/config/autoload.php file and add the item you want loaded to the autoload array
- $autoload['libraries'] = array('database','session','email','validation');
- $autoload['helper'] = array('url','form','text','date','security');
- $autoload['plugin'] = array('captcha');
- $autoload['model'] = array();
- $autoload['config'] = array();

- Text Helper
  - $string = "Here is a nice text";
  - $string = word_limiter($string, 4); // for English
- URL Helper
  - base_url()
  - anchor('news/local/123', 'My News');
- Email Helper
  - valid_email('email'); //return true/false
  - send_email('recipient', 'subject', 'message')

CodeIgniter Libraries

- Language Class
- Input and Security Class
- Pagination Class
- Session Class
- URI Class
- Email Class
- Form Validation Class

Using CodeIgniter Libraries

- $this->load->library('class name');
  - Email Class
  - $this->load->library('email');

```
$this->load->library('email');
$this->email->from('your@example.com', 'Your Name');
$this->email->to('someone@example.com');
$this->email->cc('another@another-example.com');
$this->email->bcc('them@their-example.com');
$this->email->subject('Email Test');
$this->email->message('Testing the email class.');
$this->email->send();
```

Input and Security Class

- Do not use isset
- $this->input->post('some_data');
  - The function returns FALSE (boolean) if the item you are attempting to retrieve does not exist.
- $this->input->post('some_data', TRUE);
- $this->input->get('some_data', TRUE);
- $this->input->get_post('some_data', TRUE);

Form Validation

- Setting Validation Rules
  - $this->form_validation->set_rules('username', 'Username', 'required|min_length[5]|max_length[12]');
  - $this->form_validation->set_rules('password', 'Password', 'required|matches[passconf]');
  - $this->form_validation->set_rules('passconf', 'Password Confirmation', 'required');
  - $this->form_validation->set_rules('email', 'Email', 'required|valid_email');

Form Validation Controller

- $this->form_validation->run(); return true / false
- Show error message
  - <?=validation_errors();?> // add view html
  - <?=form_error('user_name'); ?>

Exercise

- 建立新聞系統表單
- 表單內容：新聞標題，新聞內容，新聞日期，聯絡人 email
- 利用　Form Validation 驗證表單

Creating Libraries

- Your library classes should be placed within your application/libraries folder
- File names must be capitalized. For example: Myclass.php
- Class declarations must be capitalized. For example: class Myclass
- Class names and file names must match.

The Class File

```
<?php if ( ! defined('BASEPATH')) exit('No direct script
access allowed');

class Someclass {

    function display_error()

    {

    }

    function ErrMsg()

    {

    }

}

?>
```

Using Your Class

- $this->load->library('someclass');
- Once loaded you can access your class using the lower case version:
- $this->someclass->some_function(); // Object instances will always be lower case

Utilizing CodeIgniter Resources within Your Library

- $this->load->helper('url');
- $this->load->library('session');
- $this->config->item('base_url');
- $this, only works directly within your controllers, your models, or your views.

- If you would like to use CodeIgniter's classes from within your own custom classes you can do so as follows:
- First, assign the CodeIgniter object to a variable:
  - $CI =& get_instance();
    - $CI->load->helper('url');
    - $CI->load->library('session');

Exercise

- Write your own Library and edit autoload.php file to auto load it.
- Write library class system_message, and it contains ErrMsg function.
  - <script language=”javascript”>
  - alert(“hello world”);
  - history.go(-1);
  - </script>
- Add another function.

What is a Model?

- model class that contains functions to insert, update, and retrieve your data.
- Model classes are stored in your application/models/ folder.
- Example: application/models/user_model.php

```
class User_model extends Model {
    function User_model()
    {
        parent::Model();
    }
}
```

Loading a Model

- $this->load->model('Model_name');
- if you have a model located at application/models/blog/queries.php you'll load it using:
- $this->load->model('blog/queries');
- Auto-loading:
  - Edit application/config/autoload.php

Example: controller loads a model

```
class Blog extends Controller
{
    function index()
    {
      $this->load->model('Blog');
      $data['query'] = $this->Blog->get_last_ten_entries();
      $this->load->view('blog', $data);
    }
}
```

Database Configuration

- Edit: application/config/database.php
  - $db['default']['hostname'] = "localhost";
  - $db['default']['username'] = "root";
  - $db['default']['password'] = "";
  - $db['default']['database'] = "database_name";
  - $db['default']['dbdriver'] = "mysql";
  - $db['default']['dbprefix'] = "";
  - $db['default']['pconnect'] = FALSE;
  - $db['default']['cache_on'] = FALSE;

Database Configuration

- $active_group = "default";
- $active_record = TRUE;

Queries

- Original: $result = mysql_query($sql) or die (mysql_error());
- CI: $query = $this->db->query('YOUR QUERY HERE');
- $sql = "SELECT * FROM some_table WHERE id = ? AND status = ? AND author = ?";
- $this->db->query($sql, array(3, 'live', 'Rick'));
- Escaping Queries
  - $this->db->escape($title);

Generating Query Results

- Result():This function returns the query result as an array of objects, or an empty array on failure.
- single result row: $query->row();

```
$query = $this->db->query("YOUR QUERY");
if ($query->num_rows() > 0)
{
   foreach ($query->result() as $row)
   {
      echo $row->title;
      echo $row->name;
      echo $row->body;
   }
}
```

Generating Query Results

- result_array():
- single result row: $query->row_array();

```
$query = $this->db->query("YOUR QUERY");

foreach ($query->result_array() as $row)
{
   echo $row['title'];
   echo $row['name'];
   echo $row['body'];
}
```

Result Helper Functions

- $query->num_rows()
- $query->free_result()
- $this->db->insert_id()
- $this->db->count_all();
  - echo $this->db->count_all('my_table');
- $this->db->insert_string();
  - $data = array('name' => $name, 'email' => $email, 'url' => $url);
  - $str = $this->db->insert_string('table_name', $data);

Result Helper Functions

- $this->db->update_string();
  - $data = array('name' => $name, 'email' => $email, 'url' => $url);
  - $where = "author_id = 1 AND status = 'active'";
  - $str = $this->db->update_string('table_name', $data, $where);

Active Record Class

- It allows information to be retrieved, inserted, and updated in your database with minimal scripting.
- It also allows for safer queries, since the values are escaped automatically by the system

Selecting Data

- $query = $this->db->get('mytable');
  - Produces: SELECT * FROM mytable
- $query = $this->db->get('mytable', 10, 20);
  - Produces: SELECT * FROM mytable LIMIT 20, 10 (in MySQL. Other databases have slightly different syntax)

Selecting Data

- $this->db->select('title, content, date');
- $query = $this->db->get('mytable');
  - // Produces: SELECT title, content, date FROM mytable

Selecting Data

- $this->db->select_max('age', 'member_age');
  - // Produces: SELECT MAX(age) as member_age FROM members
- $this->db->select_min('age');
- $this->db->select_avg('age');
- $this->db->select_sum();

Selecting Data

- $this->db->select('title, content, date');
- $this->db->from('mytable');
- $query = $this->db->get();
  - // Produces: SELECT title, content, date FROM mytable

Select join

- $this->db->join('comments', 'comments.id = blogs.id', 'left');
  - // Produces: LEFT JOIN comments ON comments.id = blogs.id
- Options are: left, right, outer, inner, left outer, and right outer.

- $this->db->where('name', $name);
  - // Produces: WHERE name = 'Joe'
- $this->db->where('name', $name);
- $this->db->where('status', $status);
  - // WHERE name 'Joe' AND status = 'active'
- $this->db->where('name !=', $name);
- $this->db->where('id <', $id);
  - // Produces: WHERE name != 'Joe' AND id < 45
- $array = array('name !=' => $name, 'id <' => $id, 'date >' => $date);
  - $this->db->where($array);
- $where = "name='Joe' AND status='boss' OR status='active'";
- $this->db->where($where);

- $this->db->where('name !=', $name);
- $this->db->or_where('id >', $id);
  - // Produces: WHERE name != 'Joe' OR id > 50
- $names = array('Frank', 'Todd', 'James');
- $this->db->where_in('username', $names);
  - // Produces: WHERE username IN ('Frank', 'Todd', 'James')
- $names = array('Frank', 'Todd', 'James');
- $this->db->or_where_in('username', $names);
  - // Produces: OR username IN ('Frank', 'Todd', 'James')
- $this->db->where_not_in();
- $this->db->or_where_not_in();

- $this->db->like('title', 'match', 'before');
  - // Produces: WHERE title LIKE '%match'
- $this->db->like('title', 'match', 'after');
  - // Produces: WHERE title LIKE 'match%'
- $this->db->like('title', 'match', 'both');
  - // Produces: WHERE title LIKE '%match%'
- $this->db->like('title', 'match');
- $this->db->or_like('body', $match);
  - // WHERE title LIKE '%match%' OR body LIKE '%match%'
- $this->db->not_like();
- $this->db->or_not_like();

- $this->db->group_by(array("title", "date"));
  - // Produces: GROUP BY title, date
- $this->db->order_by("title", "desc");
  - // Produces: ORDER BY title DESC
- $this->db->order_by('title desc, name asc');
  - // Produces: ORDER BY title DESC, name ASC
- $this->db->order_by("title", "desc");
- $this->db->order_by("name", "asc");
  - // Produces: ORDER BY title DESC, name ASC
- $this->db->limit(10, 20);
  - // Produces: LIMIT 20, 10 (in MySQL. Other databases have slightly different syntax)

Inserting Data

- $data = array(
- 'title' => 'My title' ,
- 'name' => 'My Name' ,
- 'date' => 'My date'
-);
- $this->db->insert('mytable', $data);
  - // Produces: INSERT INTO mytable (title, name, date) VALUES ('My title', 'My name', 'My date')

Updating Data

- $data = array(
- 'title' => $title,
- 'name' => $name,
- 'date' => $date
-);
- $this->db->where('id', $id);
- $this->db->update('mytable', $data);
  - // Produces: UPDATE mytable SET title = '{$title}', name = '{$name}', date = '{$date}' WHERE id = $id

Deleting Data

- $id = array('1', '2', '3');
- $this->db->where_in('user_id', $id);
- $this->db->delete('mytable');
- // Produces: DELETE FROM mytable WHERE user_id in (1, 2, 3)

Exercise

- Write news model which contains add_news, edit_news, update_news, get_last_entries function.
- add_news($data), edit_news($data), update_news($data), get_last_entries($num)
- Wirte Controller function to insert, update and select data with news model.

Scaffolding

- It feature provides a fast and very convenient way to add, edit, or delete information in your database during development.
- To set a secret word, open your application/config/routes.php file and look for this item:
  - $route['scaffolding_trigger'] = '';

Enabling Scaffolding

- <?php
- class Blog extends Controller {
- 	function Blog()
- 	{
- 		parent::Controller();
- 		$this->load->scaffolding('table_name');
- 	}
- }
- ?>
- example.com/index.php/blog/abracadabra/

How to support it

- 繁體中文討論區
  - http://ci.wuboy.twbbs.org/forum/
- 繁體中文線上文件
  - http://ci.wuboy.twbbs.org/user_guide/
- CI Wiki
  - http://codeigniter.com/wiki/Special:Titles
- 酷學園討論區
  - http://tinyurl.com/m8vjq7
- PTT BBS PHP 討論版