

ZEND FRAMEWORK

By Virote Tangnitipong

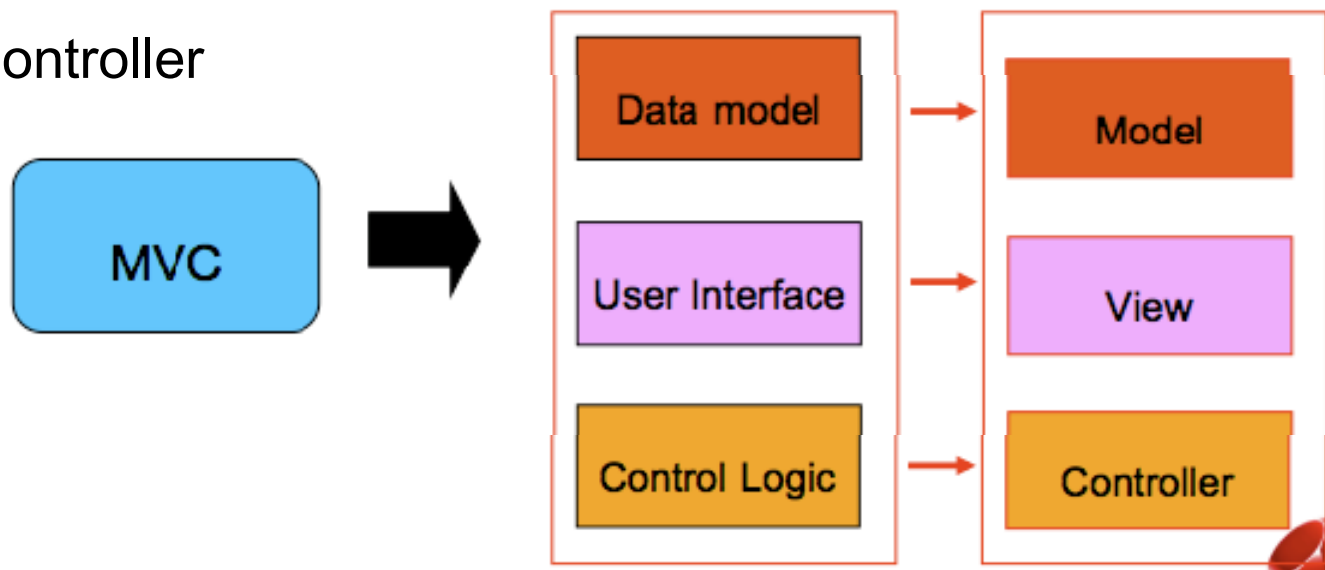
OUTLINE

- What is MVC ?
- Zend Framework ?

MVC

เป็นหนึ่งในหลายๆ **pattern** ของ **Design pattern** ซึ่งเป็น **Model** หรือรูปแบบในการวางโครงสร้างโปรแกรม โดยแยกเป็น

- Model
- View
- Controller



MVC - MODEL

จะประกอบด้วย **class** ที่เชื่อมต่อกับ **RDBMS** เพื่อช่วยจัดการงานด้าน **Database** เช่น

- ดูแลในเรื่องของการติดต่อสื่อสารระหว่าง **Object** และ **Database** โดยที่ผู้พัฒนาไม่ต้องยุ่งยากกับการใช้ **SQL command**
- เป็นงานด้านการตรวจสอบความสัมพันธ์ของข้อมูล มีผลกับฐานข้อมูล
- **Handles validation** (ตรวจสอบความถูกต้อง), **association** (ความสัมพันธ์ระหว่างฐานข้อมูล), **transactions**, and more...

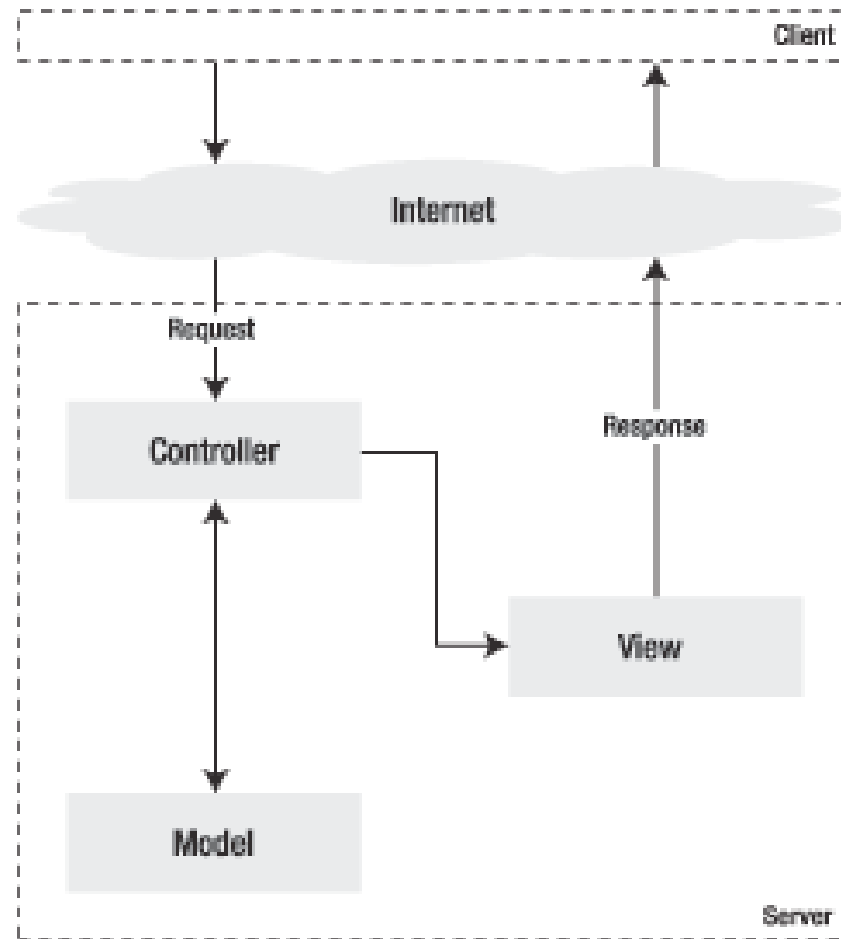
MVC - VIEW

- เป็นส่วนที่ต้องแสดงผลผ่าน **web browser**
- เขียนด้วยพื้นฐานของ **HTML(.phtml)**, แทรกด้วย **script** ของ **PHP**
- การทำงานสัมพันธ์อยู่กับ **controller** เช่น ใน **controller** มีการกำหนด **method** ที่มีชื่อว่า **index,home,contact** เป็นต้น โดยทั้ง 3 **method** ต้องมีการแสดงผลต่างกัน ดังนั้นต้องเขียนไฟล์ในการแสดงผลใน โฟลเดอร์ **view** 3 ไฟล์ ได้แก่ **index.phtml , home.phtml , contact.phtml** เป็นต้น
- สนับสนุน **Ajax**

MVC - CONTROLLER

- เป็นส่วนที่ทำงานเป็นอันดับแรกเมื่อมีโปรแกรมถูกเรียก จาก **Web browser**
- เป็นส่วนที่ติดต่อการทำงานระหว่างผู้ใช้และโปรแกรม
- มีการติดต่อกับ **Database** (ฐานข้อมูล) ด้วย **Model** และแสดงผลข้อมูลผ่านทาง **View**
- เป็นส่วนที่มีการประมวลผลหลัก ของโปรแกรม

MVC - ขั้นตอนการทำงานร่วมกัน



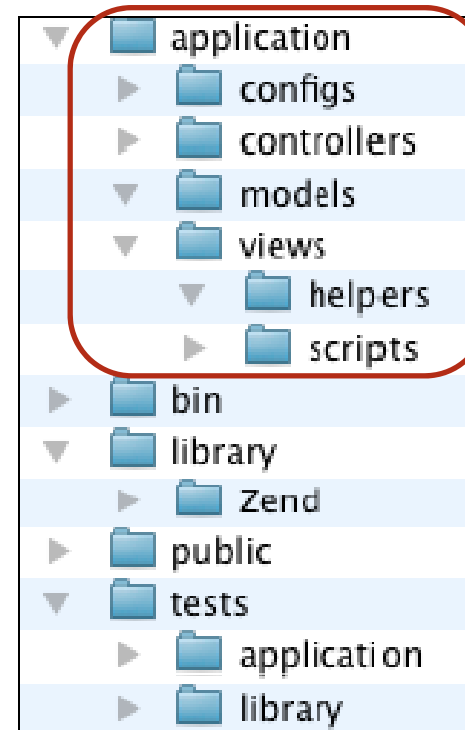
MVC – ประโยชน์

- ง่ายต่อการ เพิ่ม, แก้ไข, ลบ ส่วนเชื่อมต่อผู้ใช้
- สามารถแยกการแสดงผลได้หลายมุมมอง (สำหรับกรณีที่มีข้อมูลเหมือนกัน)
- สามารถทำการเปลี่ยนแปลง **Logic Control** ได้ง่าย
- ช่วยลดเวลาในการพัฒนาโปรแกรม โดยการ **reuse code** และการแบ่งหน้าที่กันในการเขียนโปรแกรม

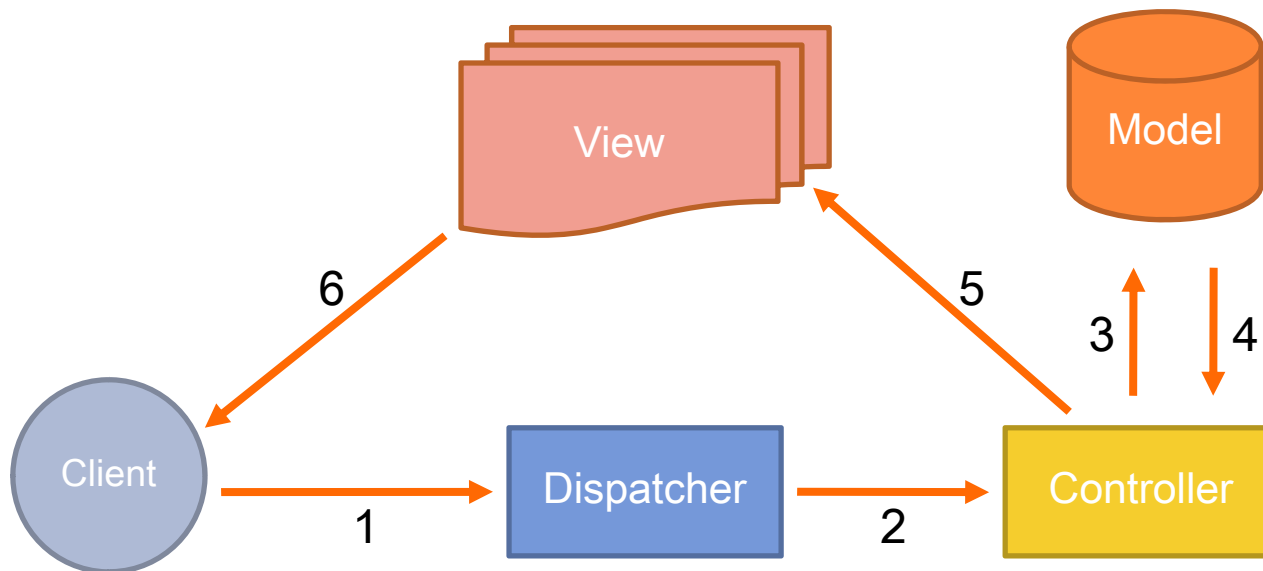
ZEND FRAMEWORK

รูปแบบในการเขียนโปรแกรมแบบหนึ่ง โดยอาศัยหลักการทำงานของ **MVC** ซึ่งจะใช้หลักพื้นฐานของ **front controller design pattern** ในการตัดสินใจที่จะเรียกใช้หรือหยุดใช้ งาน **controller/action** เมื่อมีการร้องขอผ่านทางไฟล์ **php** ต่าง ๆ และส่งต่อการควบคุมให้กับ **Action Controller** รวมทั้งการคืนค่า **response**

โดยจะมีโครงสร้าง **directory** แยกเป็น **controller, models** และ **view** อยู่ภายใน **application directory**



ZEND FRAMEWORK – ขั้นตอนการทำงานร่วมกันของ MVC



ZEND FRAMEWORK – SETTING UP THE ENVIRONMENT

- PHP 5.2.4 (ขึ้นไป) with php_pdo module
- Apache with mod_rewrite
- Zend Framework Library (current 1.10.8)
- MySql 4.0+ (option)
- Oracle Instant Client (option)

ZEND FRAMEWORK – CREATING A PROJECT

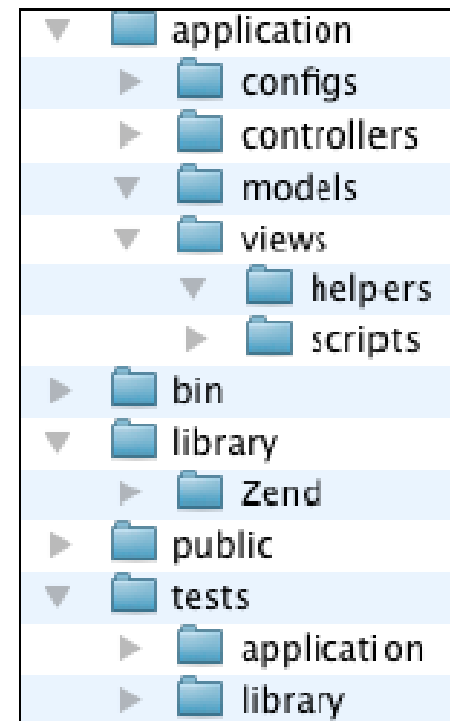
จะใช้เครื่องมือที่เรียกว่า **zend tool** ในการสร้าง **project structure**

รูปแบบคำสั่ง

- For Windows users:
bin\zf.bat create project
- For Linux and Mac users:
bin/zf.sh create project

รูปแบบคำสั่ง (อื่นๆ)

bin/zf.bat create project /my/other/path
bin/zf.sh create project /my/other/path



ZEND FRAMEWORK – CREATING A PROJECT (ต่อ)

ในการสร้าง **project structure** นั้น **zend tool** จะทำการสร้างไฟล์ให้อัตโนมัติ
คือ

- **Bootstrap file** สำหรับใช้เป็นจุดในการรับ **request** ของผู้ใช้ และกำหนดค่าเริ่มต้นของโปรแกรมผ่านทาง **Zend_Application** ซึ่งจะอยู่ในไฟล์ **index.php**
- **Action controller file** สำหรับใช้ในส่วนของการ **control logic** ของ **application** โดยจะสร้างตัว **controller** มาให้ 2 ไฟล์ คือ **IndexController.php** และ **ErrorController.php** ซึ่งจะอยู่ใน **application/controllers**

ZEND FRAMEWORK – CREATING A PROJECT (ต่อ)

ตัวอย่าง bootstrap file (index.php)

```
<?php
// Define path to application directory
defined('APPLICATION_PATH')
    || define('APPLICATION_PATH', realpath(dirname(__FILE__) .
        '/../application'));

// Define application environment
defined('APPLICATION_ENV')
    || define('APPLICATION_ENV', (getenv('APPLICATION_ENV') ?
        getenv('APPLICATION_ENV') : 'production'));

// Ensure library/ is on include_path
set_include_path(implode(PATH_SEPARATOR, array(
    realpath(APPLICATION_PATH . '/../library'),
    get_include_path(),
)));
/** Zend_Application */
require_once 'Zend/Application.php';

// Create application, bootstrap, and run
$application = new Zend_Application(APPLICATION_ENV,
    APPLICATION_PATH . '/configs/application.ini');
$application->bootstrap()->run();
```

ZEND FRAMEWORK – CREATING A PROJECT (ต่อ)

ตัวอย่าง action controller file (IndexController.php)

```
<?php
class IndexController extends Zend_Controller_Action
{
    public function init()
    {
        $this->_helper->viewRenderer->setNoRender();
    }

    public function indexAction()
    {
        $this->getResponse()
            ->appendBody('Hello from indexAction');
    }
}
```

ZEND FRAMEWORK – CREATING A PROJECT (ต่อ)

ตัวอย่าง action controller file (IndexController.php)

```
<?php
class IndexController extends Zend_Controller_Action
{
    public function init()
    {
        /* Initialize action controller here */
    }

    public function indexAction()
    {
        // action body
    }
}
```


ZEND FRAMEWORK – CREATING A PROJECT (ต่อ)

การสร้าง action อื่นๆ

รูปแบบคำสั่ง

- For Windows users:
bin\zf.bat create action about index
- For Linux and Mac users:
bin/zf.sh create action about index

ZEND FRAMEWORK – CREATING A PROJECT (ต่อ)

ตัวอย่าง action controller file (IndexController.php)

```
<?php
class IndexController extends Zend_Controller_Action
{
    public function init()
    {
        $this->_helper->viewRenderer->setNoRender();
    }

    public function indexAction()
    {
        $this->getResponse()
            ->appendBody('Hello from indexAction');
    }

    public function aboutAction()
    {
        $this->getResponse()
            ->appendBody('Hello from aboutAction');
    }
}
```

ZEND FRAMEWORK – ZEND CONTROLLER

- Front Controller
- Action Controller
- Action
- Modules
- Request Object
- Response Object
- Router
- Dispatcher
- Dispatch Loop

ZEND FRAMEWORK – ZEND CONTROLLER (ต่อ)

○ Front Controller

- จัดการกับการ requests
- ส่งต่อการควบคุมให้กับ Action Controller
- คืนค่า response

○ Action Controller

- ในการสร้าง Action Controller เราจะสร้าง class ซึ่ง extends Zend_Controller_Action
- ในการตั้งชื่อ class เราจะตั้งชื่อให้ลงท้ายว่า Controller ยกตัวอย่างเช่น IndexController
- ตัวอักษร _ ในชื่อ class จะแทน subdirectory อย่างเช่น Test_MyController จะหมายถึง Test/MyController.php
- ใน Action Controller เราสามารถสร้าง method เพื่อ listen และแทรกคำสั่งใน event ต่อไปนี้
 - init(): method ที่จะถูกเรียกเมื่อ object ของ Action Controller ถูกสร้างขึ้น
 - preDispatch(): method ที่จะถูกเรียกก่อนการ dispatch action
 - postDispatch(): method ที่จะถูกเรียกหลังการ dispatch action

ZEND FRAMEWORK – ZEND CONTROLLER (ต่อ)

○ Action Controller (ต่อ)

• Utility Methods

- `_forward` : ใช้เพื่อย้ายไปยัง **action** อื่น
- `_redirect` : ใช้เพื่อย้ายไปยัง **url** อื่น
- `render` : ใช้เพื่อเปลี่ยนไปใช้ **view** อื่นแทน **default view**
- `__call` : สามารถ **override method** นี้เพื่อควบคุมการทำงาน **undefined action** ว่าจะให้ **forward** ไปยัง **action** ไหน

○ Action

คือ **method** ที่ **controller** นำมาใช้งาน โดยจะต้องเป็น **public method** ที่มีชื่อลงท้ายว่า **Action** เช่น `indexAction()`, `viewAction()`

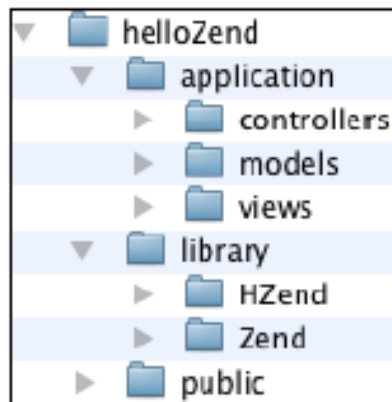
ZEND FRAMEWORK – ZEND CONTROLLER (ต่อ)

○ Modules

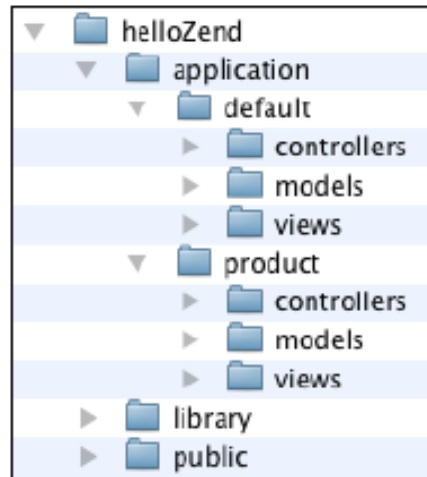
คือกลุ่มของ **action controllers, models** และ **views** ที่สัมพันธ์กัน โครงสร้าง **directory** จะสัมพันธ์กับโครงสร้าง **application directory** ชื่อคลาสของ **Controller** ควรจะนำหน้าด้วยชื่อ **Module**

ยกตัวอย่างเช่น **Cartoon_DoraemonController.php** ซึ่งจะมีโครงสร้าง **directory** เป็น **Cartoon/controller/DoraemonController.php**

ชื่อของ **Module** จะต้องเป็น **camel case** เช่นเดียวกับชื่อของ **Controller**



Basic directory



Using modules

ZEND FRAMEWORK – ZEND CONTROLLER (ต่อ)

○ Request Object

จะบรรจุข้อมูลเกี่ยวกับการ **request** เอาไว้ทั้งหมด

เช่น

- `getModuleName()` and `setModuleName()`
- `getControllerName()` and `setControllerName()`
- `getActionName()` and `setActionName()`
- `isDispatched()` and `setDispatched()`
- `getParam()` and `setParam()`
- `getParams()` and `setParams()`
- `getUserParams()` and `getUserParam()`

ตัวอย่าง

```
$front = Zend_Controller_Front::getInstance();  
$request = $front->getRequest();
```

ZEND FRAMEWORK – ZEND CONTROLLER (ต่อ)

○ Response Object

จะบรรจุข้อมูล **response** และมีความสามารถในการส่งข้อมูลกลับไปให้ **user**

เช่น

- `append (string $name, string $content)`
- `appendBody (string $content, null|string $name = null)`
- `prepend (string $name, string $content)`
- `insert (string $name, string $content, string $parent = null, boolean $before = false)`

ตัวอย่าง

```
$response = new Zend_Controller_Response_Http();  
$response->setBody('<h1>Default Body</h1>');  
$response->setBody('<p>More body</p>', 'test');  
$response->appendBody('<p>append to test segment</p>', 'test');  
$response->prepend('header', '<html><body>');  
$response->append('footer', '</body></html>');  
$response->insert('extra', '<h2>Extra body</h2>', 'default', false);  
$response->insert('more', '<p>Before footer</p>', 'footer', true);
```


ZEND FRAMEWORK – ZEND CONTROLLER (ต่อ)

○ Router

จะทำการแยกข้อมูลการ **request** เช่น **url** ออกเป็น **token** เพื่อใช้ในการตัดสินใจที่จะเรียกใช้ **module**, **controller** และ **action** ใดๆ โดยการเรียงลำดับของเส้นทางจะเรียงจาก **module**, **controller** และ **action**

เช่น

`http://domain.com/moduleName/controllerName/actionName`

ตัวอย่าง

URI	Module / Controller / Action Called
<code>http://127.0.0.1:8080/</code>	Default / IndexController / indexAction
<code>http://127.0.0.1:8080/index</code>	Default / IndexController / indexAction
<code>http://127.0.0.1:8080/index/index</code>	Default / IndexController / indexAction
<code>http://127.0.0.1:8080/index/about</code>	Default / IndexController / aboutAction

ZEND FRAMEWORK – ZEND CONTROLLER (ต่อ)

○ Router (ต่อ)

นอกจากนี้เรายังสามารถส่งข้อมูลไปยัง **controller** และ **action** ได้อีกด้วย โดยการส่งข้อมูลแบบ **GET method**

เช่น

- <http://domain.com/controller/action/key1/value1/key2/value2>
- <http://domain.com/module/controller/action/key1/value1/key2/value2>

ตัวอย่าง

- <http://127.0.0.1/index/about/name/keith>
- <http://127.0.0.1/index/about/name/keith?age=26&country=England>
- <http://127.0.0.1/index/about/name/keith/age/26/country/England>

ZEND FRAMEWORK – ZEND CONTROLLER (ต่อ)

○ Dispatcher

นำเอา **token** ที่ได้จากการ **routing** มาเป็นข้อมูลในการเลือก **class method** ของ **action controller**, จากนั้นจึงสั่งให้มันทำงานจะบรรจุข้อมูล **response** และมีความสามารถในการส่งข้อมูลกลับไปให้ **user**

○ Dispatch Loop

- `Zend_Controller_Front::getInstance()->dispatch();` จัดการกับ **request** ที่เข้ามา
- สร้าง **object request** และ **object response** ถ้ายังไม่ได้ถูกสร้างขึ้น
- ทำการ **route request**
- เข้าสู่ **dispatch loop**
 - ในการ **dispatch action** ประกอบด้วยการสร้าง **object controller** และการเรียก **method** ของ **controller**
 - ซึ่งจะทำการ **dispatch** ไปเรื่อยๆ จนกว่า **object request** จะรายงานกลับมาว่าไม่มี **action** ที่จะต้อง **dispatch** อีก
- คืนค่า **response**

ZEND FRAMEWORK – VIEW

○ Zend_View

คลาส **Zend_View** ใช้ ภาษา **PHP** เป็น **template language** สามารถกำหนดค่าตัวแปร **view** ราวกับว่ามันเป็นสมาชิกของ **object** อย่างเช่น

```
$view->varname = $varname;
```

ใช้งานตัวแปร **view** ใน **view script** ได้จากออบเจกต์ **\$this** ตัวอย่างเช่น

```
<?= $this->varname ?>
```

○ View Helper

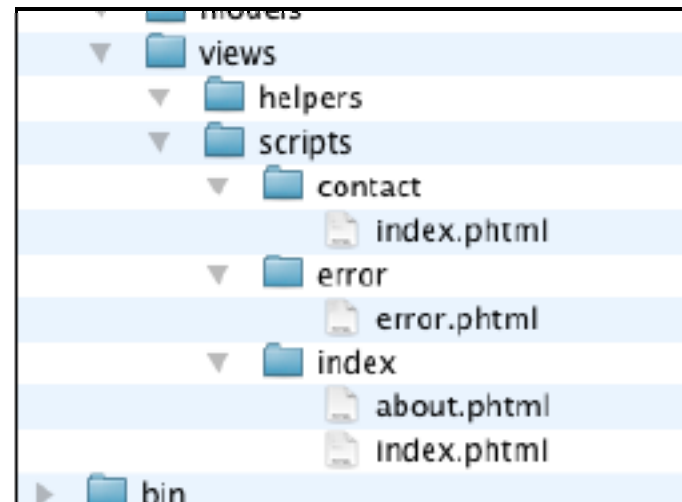
เป็นคลาสซึ่งขยายความสามารถให้กับ **Zend_View** ในการสร้าง **view** ของเราเอง เช่น

- **Action view** : สำหรับเรียกใช้ **action**
- **Doctype** : สำหรับ **return** ค่า **doctype**
- **JSON** : สำหรับการแสดงผลแบบ **JSON**
- **Various Form Element Helpers** : สำหรับการสร้าง **form**

ZEND FRAMEWORK – VIEW (view)

application/views/scripts/index/index.phtml

```
<html>
<head>
  <title>Hello Zend</title>
</head>
<body>
  <h1>Hello Zend</h1>
  <p>Hello from Zend Framework</p>
</body>
</html>
```



ZEND FRAMEWORK – VIEW (หน้า)

application/controllers/IndexController.php

```
public function init()
{

public function indexAction()
{
    $this->view->name = $this->_getParam('name', 'guest');
}
```

application/views/scripts/index/index.phtml

```
<head>
    <title>Hello Zend</title>
</head>
<body>
    <h1>Hello Zend</h1>
    <p>Hello <?=$this->Escape($this->name);?> from Zend Framework</p>
</body>
```

วิธีเรียกใช้ : <http://127.0.0.1:8080/index/index/name/keith>

เอกสารอ้างอิง

MVC

- <http://wiki.nectec.or.th/setec/Knowledge/MVC>
- <http://www.cakephp.in.th/index.php?topic=3.0>

Zend Framework

- <http://framework.zend.com/>
- <http://bunyam.com/bw/?name=knowledge&file=readknowledge&id=260>
- <http://www.phpzealots.com/node/47>
- <http://www.phpeveryday.com/pack/Zend-Framework-Intro-Step-By-Step-Tutorial>