

Confessions of an RPG Programmer: Why use Zend Framework?

Mike Pavlak

Solutions Consultant, Zend

Alan Seiden

Consultant/developer, Strategic Business Systems

June 17, 2009

About Strategic Business Systems, Inc.

- **IBM partner since 1982**
 - IBM i (AS/400) hardware, software development, consulting
 - Concentration in food & beverage and automotive industries
 - HQ in northern New Jersey
- **Zend (“the PHP company”) partner since 2008**
 - PHP’s been our preferred web technology for ourselves and clients since 2005
 - In addition to our consulting/development services, we offer Zend’s training and software
 - We represent Zend in the northeastern USA

We'll be covering...

- **What Zend Framework is**
- **Why ZF is a great match for the IBM i**
- **Intro to key concepts**
- **What ZF can do for your PHP/i projects**
- **How to get started!**

What Zend Framework is

- **A free, open source PHP framework**
- **A starting point for your PHP applications, providing**
 - Modular design
 - Security features
- **A collection of over 70 PHP components to simplify common tasks, including some for:**
 - Form creation (and reuse)
 - Logging
 - Database access
- **A demonstration of PHP 5 best practices**
- **It provides standards and great functionality but will not cramp your style. Your development is not limited in any way**

Why ZF's time is right

- **PHP is being used for critical apps on IBM i**
- **Managers, CIOs, technology architects are taking notice**
- **It's time for professional practices**
 - Standards and consistency
 - Awareness of security
 - Reuse and easy maintenance of code
 - Leverage your software investments
 - Training and support
 - Doing it “right”
- **ZF gets you there—“Enterprise PHP”—faster—and keeps you in control**

Why I use it

- **As I learn what it can do, the less boring code I write**
 - I can write less “plumbing” code
- **Use ZF’s code however you like**
 - <http://framework.zend.com/license>
 - Safe for corporate use
- **It keeps up with trends and APIs**
 - Compatibility with diverse database systems, and APIs (authentication, web services, more)

Community

- **Contributors include individuals and companies.**
Companies include:
 - Zend (of course)
 - IBM
 - OmniTI
- **Technology partners:**
 - Adobe, Google, IBM, Microsoft, nirvanix, Strikelron

Here's why ZF reminds me of the i5 world

- **Appreciation of standards: naming, parameter lists**
- **The tools you need are already integrated**
 - Common components (template system, emailer, etc.) are there for you; no need to research/download/install
 - Upgrades like a “cume tape”—all components upgraded as a well tested unit
- **ZF support available from Zend**
 - Similar to phoning IBM about i5/OS

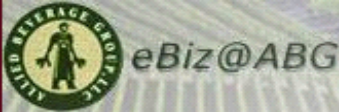



ZF's birth, early years, and maturity on i5

- **2005: PHP Collaboration Project at ZendCon**
 - Started as collection of components but coalesced
 - PHP 5, object oriented (OO) from the start
 - Set example of OO design patterns and practices
 - More on OO later
- **2007-2009: Fast progress**
 - July 2007: GA version 1.0
 - Feb. 2009: version 1.70 with db2/i5 support
 - June 2009: version 1.82; minor releases every couple of weeks
- **April 2009: ZF/i application won COMMON's "best web solution"**

COMMON award winner

Allied Beverage Group: Wine catalog/ordering system on IBM i

Welcome, Angel Wang #102

Home > [Search Product Catalog](#) > Search Results (7 products found), order for EUROPA LIQUORS (001588) PRICING MONTH: **June 2009**

[Advanced Search](#)

Search for item: [?] [\(proof °\)](#)
 with bottle price [?] to [?] (optional)

AMERICO'S INC (001588) Off Premise Terms: NET
 DBA EUROPA LIQUORS Total List: \$218.54
 155-57 PACIFIC ST, NEWARK NJ, 07102 Total Disc: \$25.00
 Phone: 973-589-8195 Est Total Net: \$ 193.54

Cases: 1 Bottles: 0

Results for **Keywords: CHARD NAPA 375ML**

Product	Code	Size	Pack	Qty	Cs/Bt	Add Items	Price		Inventory		Vintage	Info	\$ Best Buy
							\$ Case	\$ Bottle	Case	Bottle			
Acacia Chardonnay A By Acacia	5607061	375 ML	12	<input type="text" value="3"/>	<input type="button" value="cases"/>	<input type="button" value="Add"/>			26	5	NV		
Cakebread Cellars Chardonnay Napa Valley 07	7433065	375 ML	12	<input type="text" value="1"/>	<input type="button" value="bottles"/>	<input type="button" value="Add"/>			9	5	2007		
Grgich Hills Cellar Chardonnay	5544265	375 ML	12	<input type="text"/>	<input type="button" value="cases"/>	<input type="button" value="Add"/>			5	7	2006		
Levendi Chardonnay Red Hen 05	4591060	375 ML	12	<input type="text"/>	<input type="button" value="cases"/>	<input type="button" value="Add"/>			13	2	2005		
Merrvale Chardonnay Starmont 07	4223069	375 ML	12	<input type="text"/>	<input type="button" value="cases"/>	<input type="button" value="Add"/>			8	8	2007		
Schramsberg Blanc de Blanc	4056064	375 ML	12	<input type="text"/>	<input type="button" value="cases"/>	<input type="button" value="Add"/>			17	9	2005		

Instant Intro to Object Orientation (2 slides!)

Object Orientation (OO)

Here is an incredibly quick summary of OO, which you'll see used throughout ZF

OO Concept	Analogy in i5	Example
Property	a field in a data structure	<code>\$_orderNum</code>
Method	function or subprocedure	<code>isOrder()</code>
Class	Imagine an intelligent data structure containing both data (properties) and programming logic (methods), which are both called "members" of the class	<pre>class Order { protected \$_orderNum; function isOrder() { . . . } . . . }</pre>

OO Syntax

- **The arrow (->) lets you access the members (methods and properties) of an object instance**
 - `$controller = $this->getRequest()->getControllerName();`
- **Sometimes you'll also see the double colon (::), which is similar, but is used when a member is "static" (one per class)**
 - `echo Zend_Registry::get('user');`
- **If you can read this notation, you can read ZF code. You will learn to appreciate its simplicity.**

Timesavers

- **Autoloader**

- PEAR convention for class/file names

- Example: `Search_Product = Search/Product.php`

- Put this in bootstrap file:

```
require_once 'Zend/Loader/Autoloader.php';  
$loader = Zend_Loader_Autoloader::getInstance() ->  
    setFallbackAutoloader(true);
```

- Now you won't need an "include" statement to do:

```
$prod = new Search_Product();
```

- **Fluent interface**

```
$select = $db->select()  
    ->from( ...specify table and columns... )  
    ->where( ...specify search criteria... )  
    ->order( ...specify sorting criteria... );
```

Model-View- Controller Pattern

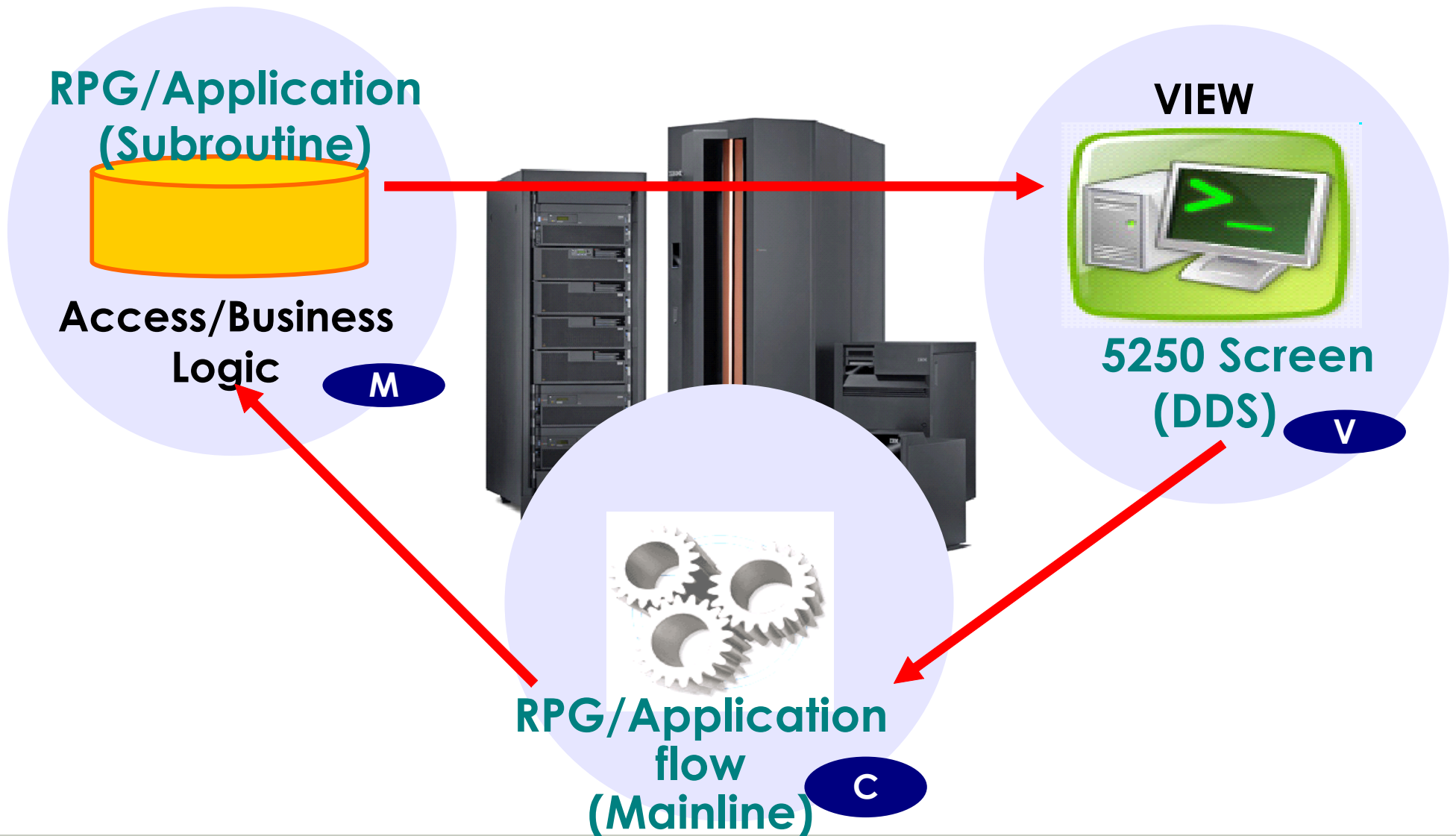
Model – View – Controller (MVC) design pattern

- **You already know this pattern from RPG/DDS**
- **With green screens, IBM handles it under the covers, so you take it for granted**
- **On the web, you must define your application's structure more explicitly**
- **Be patient...MVC seems strange at first, but you'll soon realize that you've been here before...**

MVC in detail

- **Model**
 - Reusable classes that access these resources:
 - Data
 - Business rules
 - Keep SQL and application details in one place
- **View**
 - Templates containing HTML or other output, with small bits of PHP
 - Plunk your HTML into a “view” without worrying about overwriting your mainline PHP code—helps web designers work with business programmers
- **Controller (action controller)**
 - Application flow
 - Connects model and view
 - Don't confuse with “front controller,” which just initializes the MVC
- Next: MVC from an RPG perspective

RPG Model View Controller (MVC)

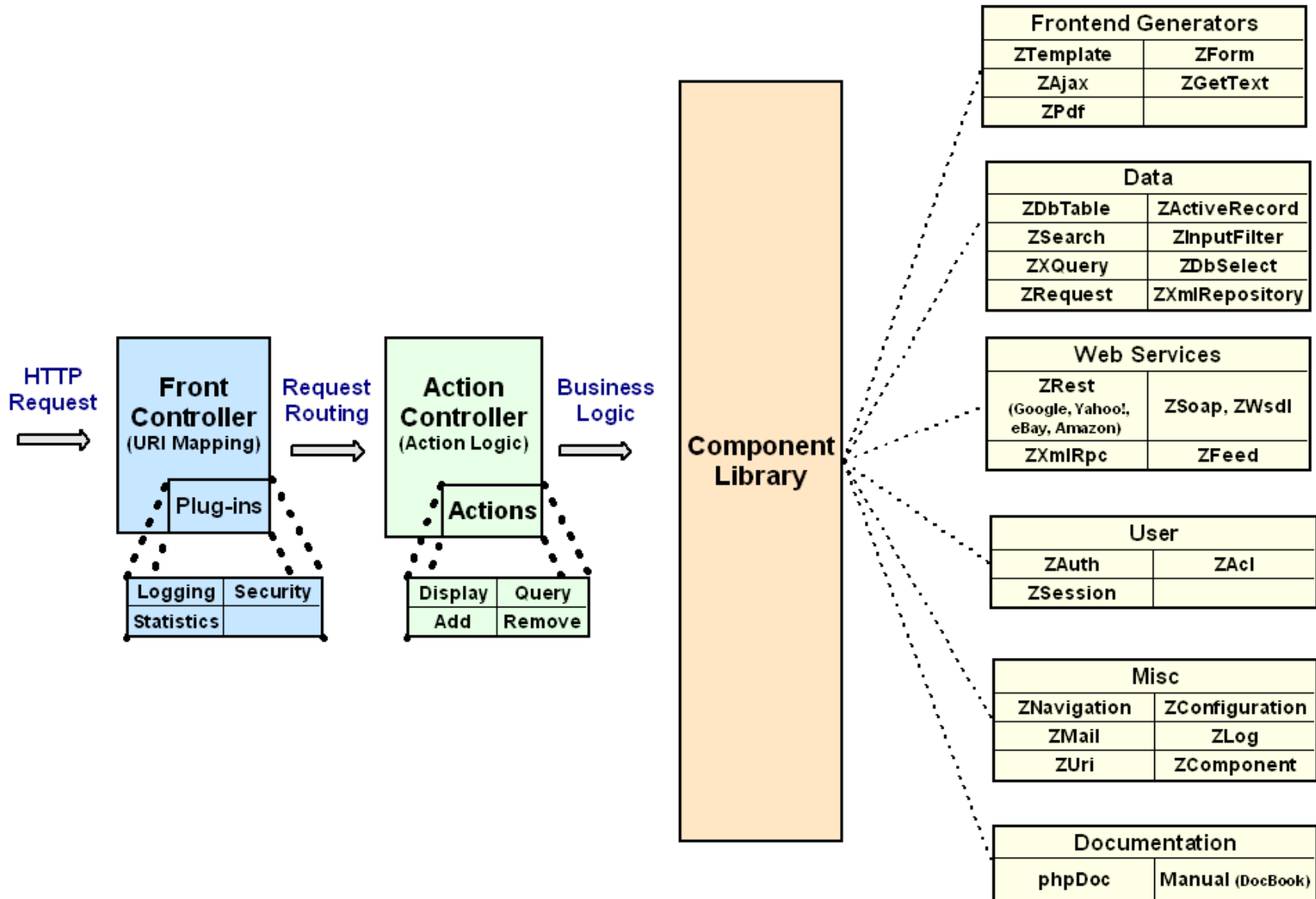


Confession

- **For my first attempt with ZF, I put all my SQL in the controller**
- **It gave me a feeling of accomplishment**
- **The MVC police did not appear**
- **Later, I moved the SQL into a model class**
 - Simplified the controller, which was getting complex and hard to understand
 - Made the SQL reusable

Initialize MVC

Front controller to action controller



Front controller Routes “friendly” URL request

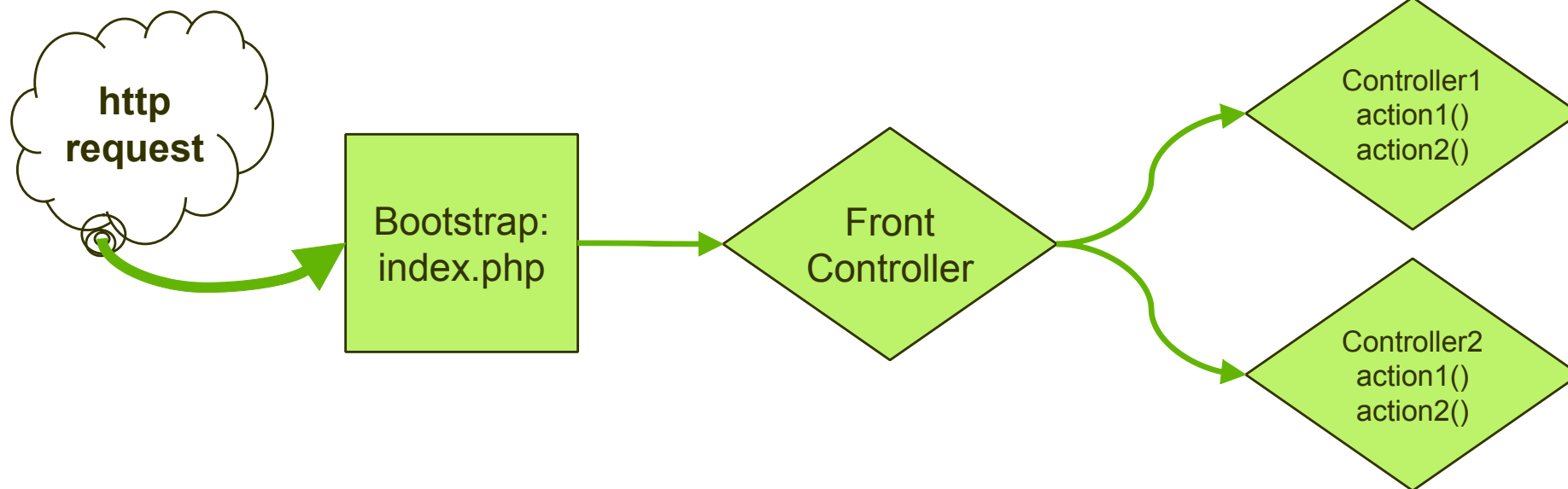
- **Default routing convention:**

- `http://example.com/controller/action/param1/value1...`

Controller maps to class name

Action maps to method name

Param/value pairs are passed to action

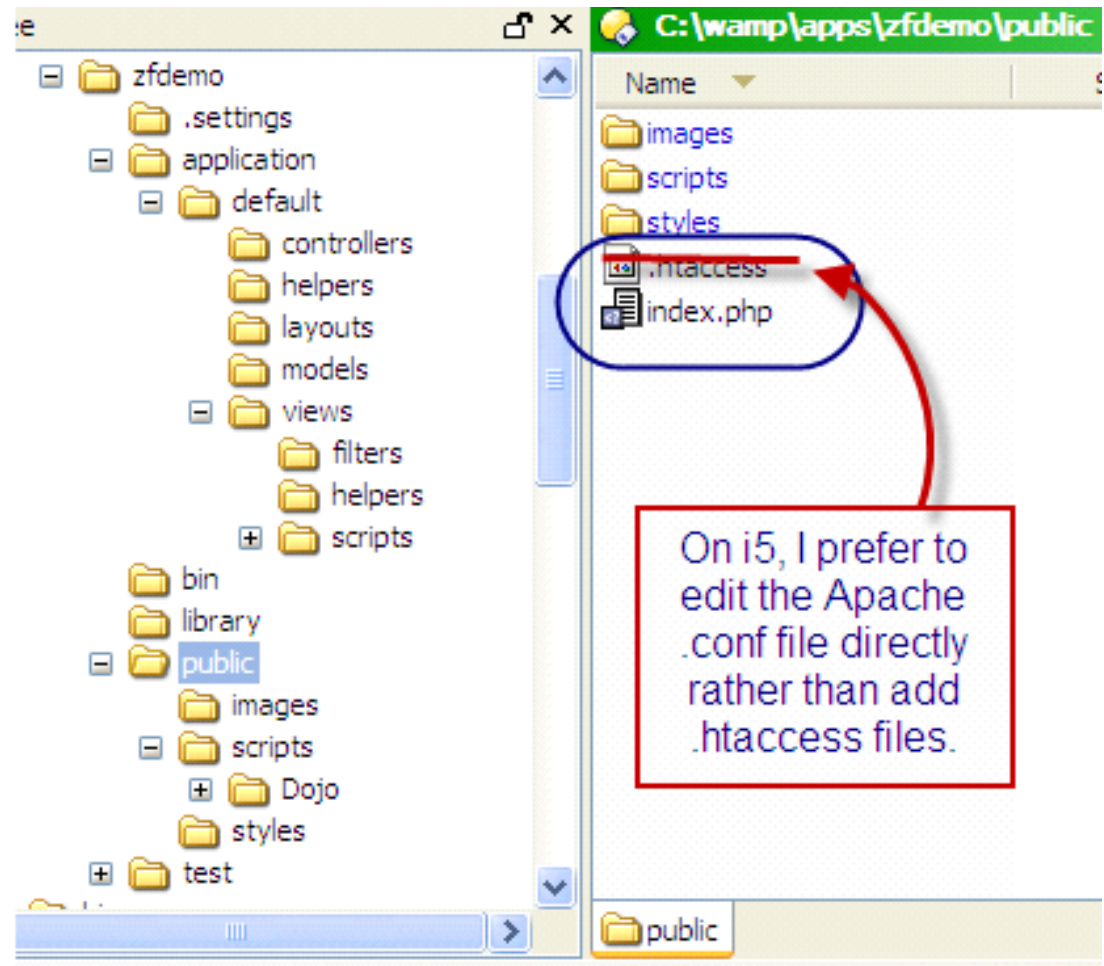


All requests routed through index.php in doc root

Document root is the only public folder.

index.php:

- initializes application
- instantiates Front Controller



Apache configuration

Most tutorials suggest `.htaccess`, but I prefer to use the main PASE Apache config file (without proxy):

```
/usr/local/Zend/apache2/conf/httpd.conf
```

```
Listen 8000
```

```
RewriteEngine on
```

```
NameVirtualHost 10.11.12.13:8000
```

```
<VirtualHost 10.11.12.13:8000>
```

```
    DocumentRoot /www/ebiz/htdocs/html
```

```
</VirtualHost>
```

```
<Directory /www/ebiz/htdocs/html/>
```

```
    # disallow .htaccess, so webserver won't search for them
```

```
    AllowOverride None
```

```
    # funnel all requests to index.php
```

```
    # except requests for static resources
```

```
    RewriteEngine On
```

```
    RewriteRule !\.(js|ico|gif|jpg|png|css|html)$ index.php
```

```
</Directory>
```

Front controller bootstrap file: index.php

```
<?php
// minimum bootstrap file (can be many variations)

// explicit, full paths save the i5 time searching for files
$paths = array(
    realpath(dirname(__FILE__) . '/../library'),
    realpath(dirname(__FILE__) . '/../application'),
    realpath(dirname(__FILE__) . '/../application/models'),
    '.'
);
set_include_path(implode(PATH_SEPARATOR, $paths));

// Prepare the front controller
$frontController = Zend_Controller_Front::getInstance();

// Dispatch the request using the front controller
$frontController->dispatch();
```

Action Controller

Action Controller

- **Controller classes handle groups of request URLs**

`http://example.com/controller/action`

Default: `IndexController`

- Organizes and groups functionality
- One class (extending `Zend_Controller_Action`) for each controller

- **Action methods in each controller class handle requests**

`http://example.com/controller/action`

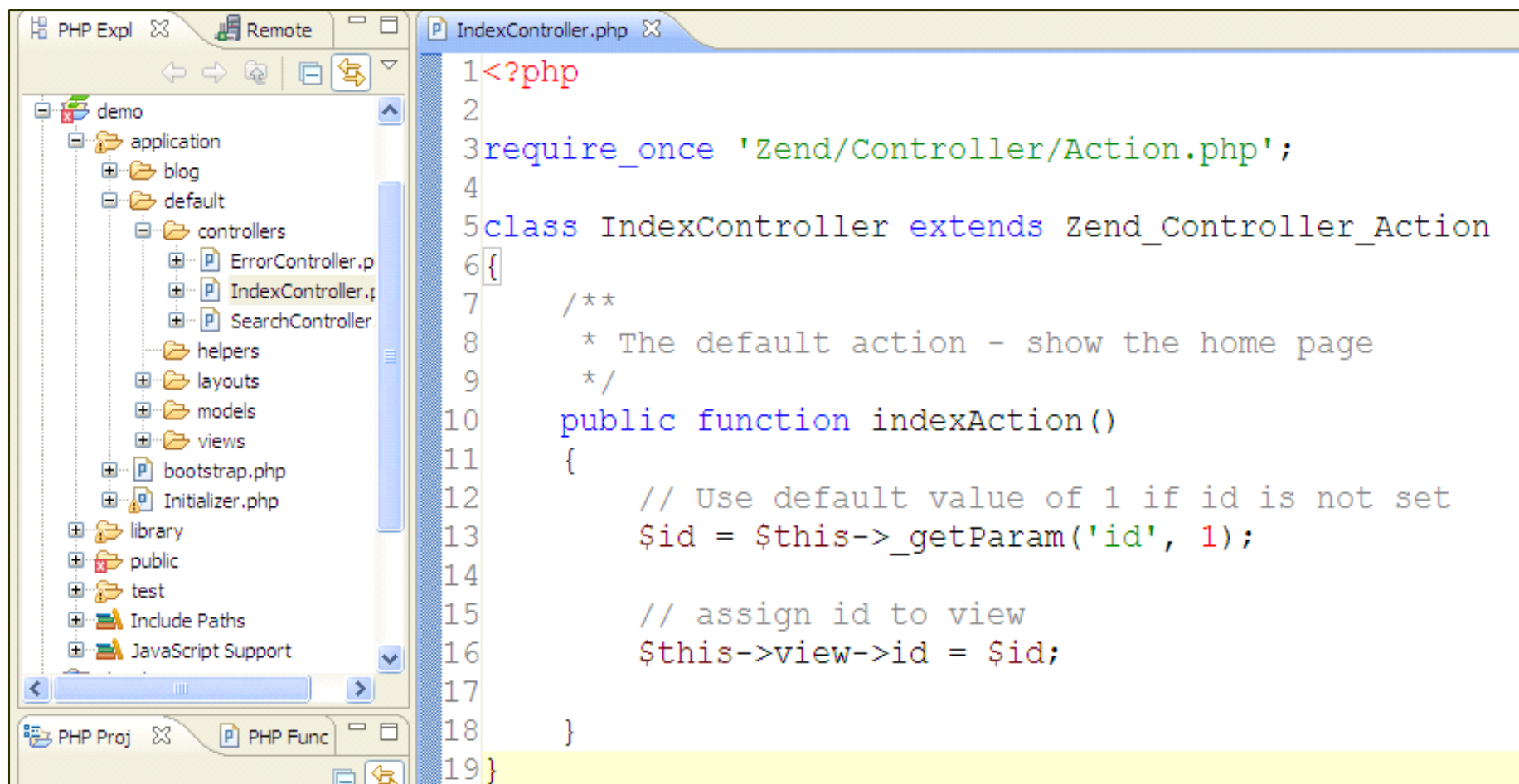
Default: `indexAction()`

- **Named like `actionAction()`**
 - **Example: If `action` is “edit” then method is `editAction()`**

More controller functionality

- **Several standard methods help organize and control the flow**
 - `init()` – called by the constructor
 - `preDispatch()` – called before the action's method
 - `postDispatch()` – called after the action's method
- **Utility methods**
 - `forward()`, `redirect()`, `getParam()`, `getRequest()`, `getResponse()`, `render()`
- **Action helpers add functionality**
 - Built-in helpers. Example: `gotoSimple()`
 - Your own helpers
 - Avoids the need to build your own base controller class

Controller example



```
1 <?php
2
3 require_once 'Zend/Controller/Action.php';
4
5 class IndexController extends Zend_Controller_Action
6 {
7     /**
8      * The default action - show the home page
9      */
10    public function indexAction()
11    {
12        // Use default value of 1 if id is not set
13        $id = $this->_getParam('id', 1);
14
15        // assign id to view
16        $this->view->id = $id;
17
18    }
19 }
```

View

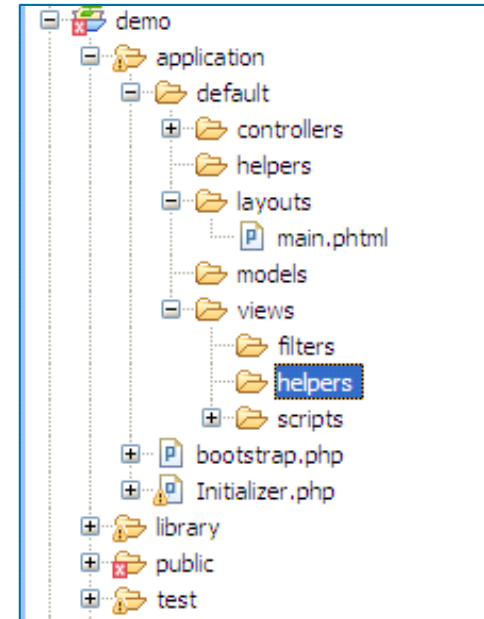
- **Scripts (templates)**
 - PHP-based script templates to present data
 - Should contain only display logic, not business logic
 - Default naming: “myaction.phtml”
- **Helpers**
 - Classes and methods that provide reusable view functionality
 - Examples of built in view helpers: `escape()`, `formText()`, `partial()`, `partialLoop()`, `headTitle()`
 - Write your own, too
- **Layout**
- **Placeholders**

What View means to you

- **You can plunk HTML right into the view script and replace literals with PHP echo statements:**
 - `<?php echo $this->productNum ?>`
- **ZF provides smart defaults**
 - The `$this->escape()` view helper uses PHP's `htmlspecialchars()` function, recommended by most security experts.

My own view helper: TitleCase.php

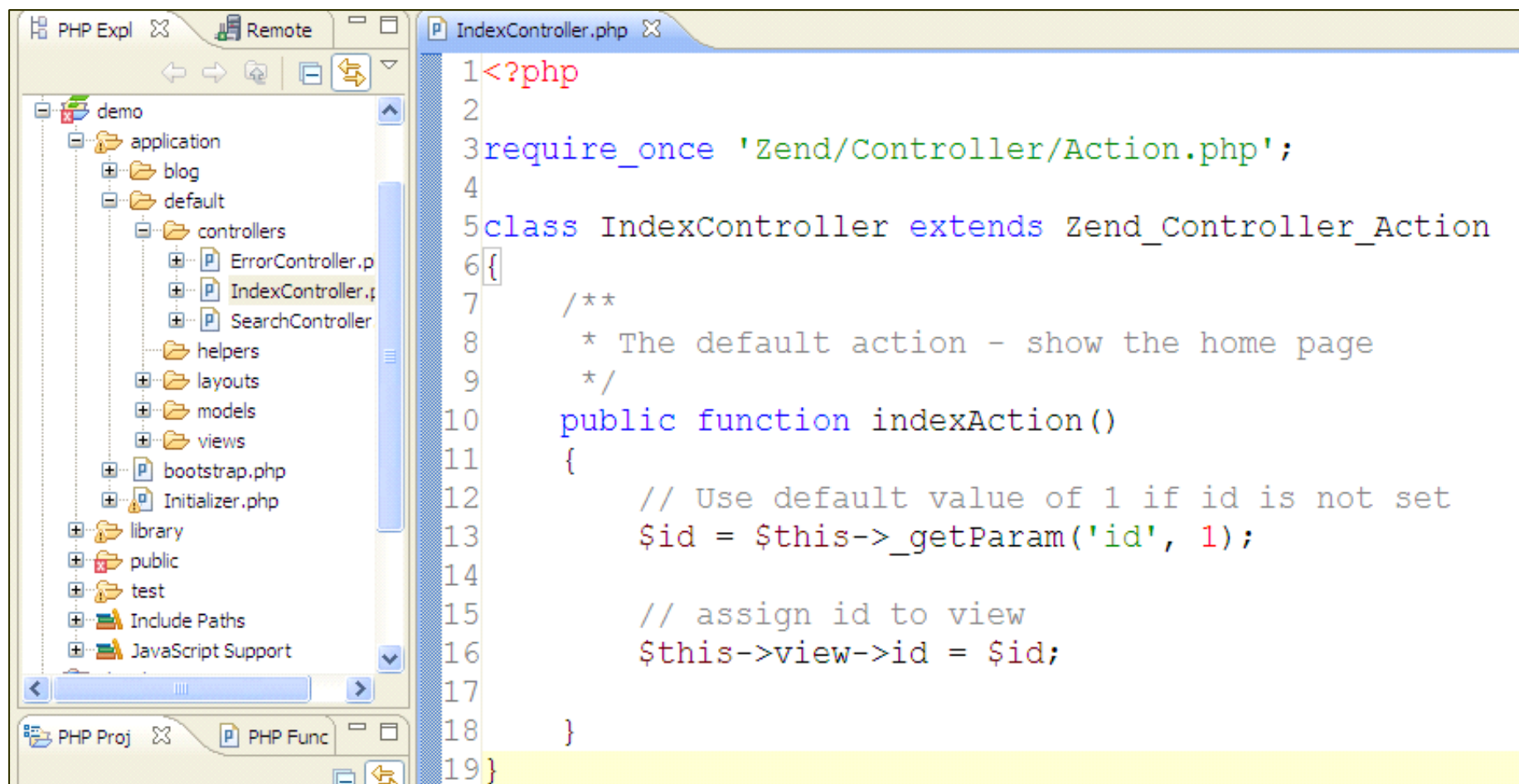
```
class Zend_View_Helper_Title_Case {  
  
    public $view;  
  
    public function titleCase($string = '')  
    {  
        return ucwords(strtolower(trim($string)));  
    } //(public function titleCase())  
  
    public function setView(Zend_View_Interface $view) {  
        $this->view = $view;  
    }  
}
```



Usage:

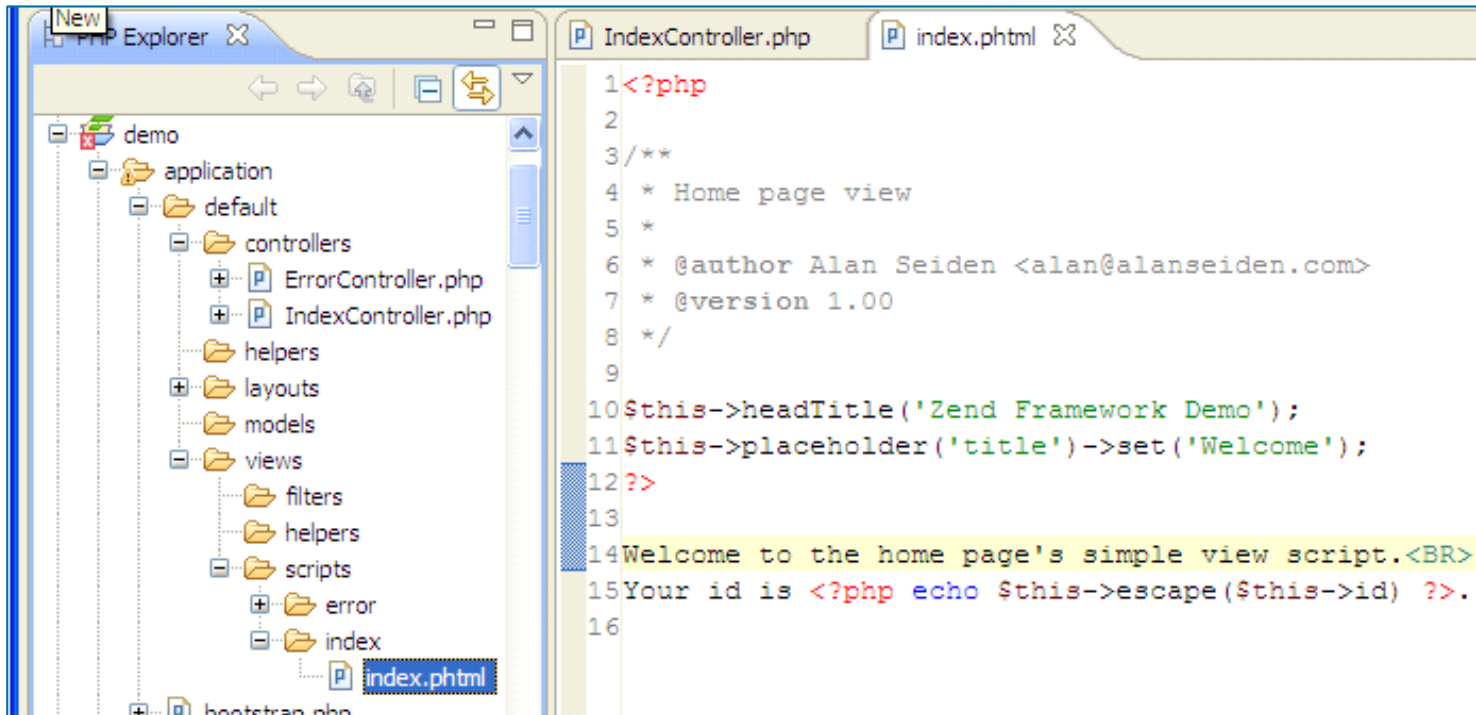
```
echo $this->titleCase('mozilla  
firefox');  
  
// Mozilla Firefox
```

Controller (again)...leads to view

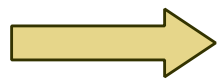


```
1 <?php
2
3 require_once 'Zend/Controller/Action.php';
4
5 class IndexController extends Zend_Controller_Action
6 {
7     /**
8      * The default action - show the home page
9      */
10    public function indexAction()
11    {
12        // Use default value of 1 if id is not set
13        $id = $this->getParam('id', 1);
14
15        // assign id to view
16        $this->view->id = $id;
17
18    }
19 }
```

View script automatically rendered



The screenshot shows a PHP Explorer window on the left with a tree view of a project named 'demo'. The tree view includes folders for 'application', 'default', 'controllers', 'helpers', 'layouts', 'models', 'views', 'filters', 'scripts', 'error', and 'index'. The 'index' folder contains 'index.phtml'. The main editor window shows the code for 'IndexController.php' and 'index.phtml'. The code in 'IndexController.php' includes a PHP header, a comment block, and two lines of code: `$this->headTitle('Zend Framework Demo');` and `$this->placeholder('title')->set('Welcome');`. The code in 'index.phtml' includes a PHP header, a comment block, and two lines of code: `14Welcome to the home page's simple view script.
` and `15Your id is <?php echo $this->escape($this->id) ?>.`



Zend Framework Demo

http://localhost/zfdemo/

Welcome

Welcome the home page's simple view script.
Your id is 1.

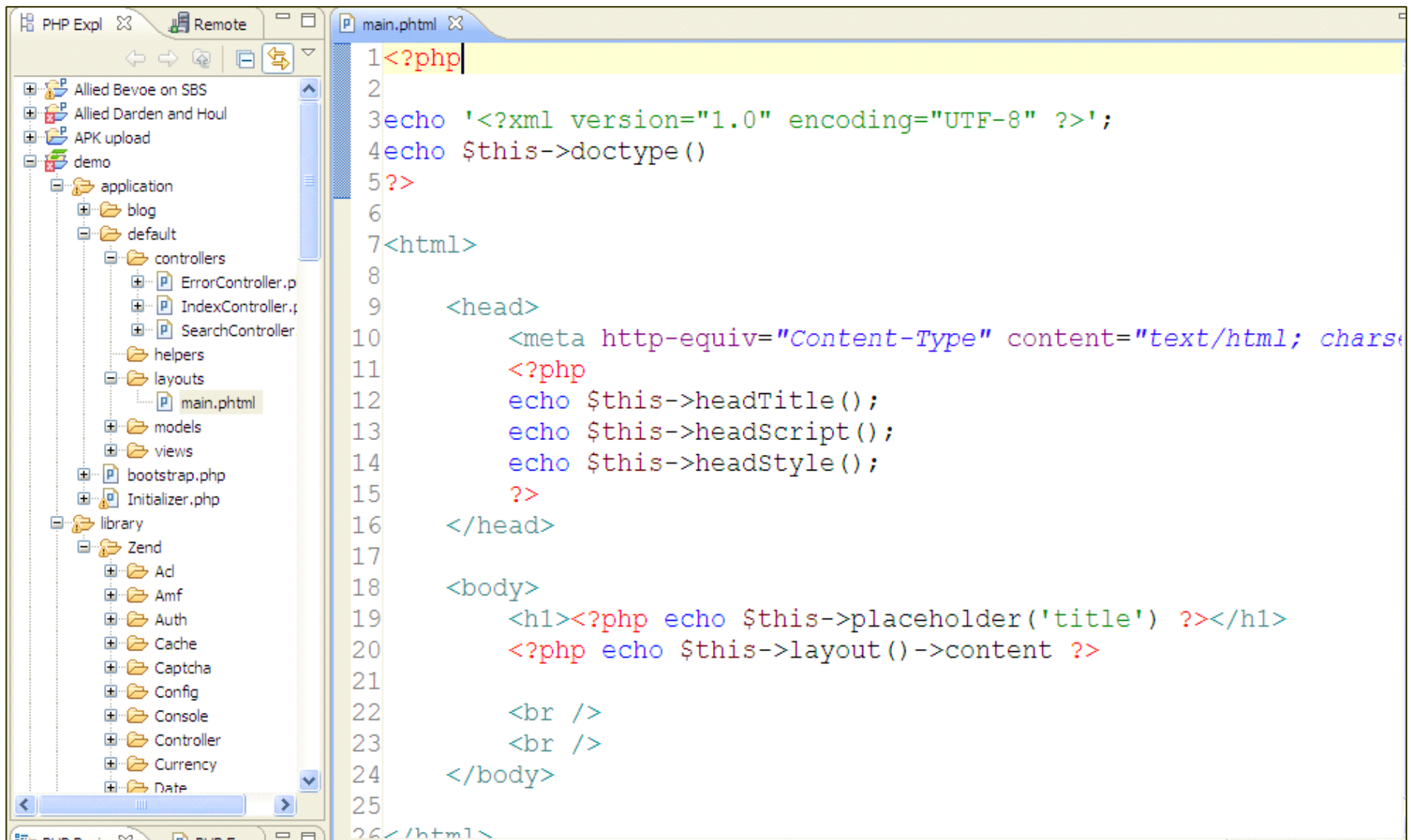
Zend Framework Demo

http://localhost/zfdemo/index/index/id/39

Welcome

Welcome to the home page's simple view script.
Your id is 39.

Zend_Layout



The image shows a code editor window with two panes. The left pane displays a file tree for a project named 'demo'. The tree structure is as follows:

- demo
 - application
 - blog
 - default
 - controllers
 - ErrorController.p
 - IndexController.p
 - SearchController
 - helpers
 - layouts
 - main.phtml
 - models
 - views
 - bootstrap.php
 - Initializer.php
 - library
 - Zend
 - Ad
 - Amf
 - Auth
 - Cache
 - Captcha
 - Config
 - Console
 - Controller
 - Currency
 - Date

The right pane shows the content of the file 'main.phtml', with line numbers 1 through 26. The code is as follows:

```
1 <?php
2
3 echo '<?xml version="1.0" encoding="UTF-8" ?>';
4 echo $this->doctype()
5 ?>
6
7 <html>
8
9     <head>
10         <meta http-equiv="Content-Type" content="text/html; charse
11         <?php
12         echo $this->headTitle();
13         echo $this->headScript();
14         echo $this->headStyle();
15         ?>
16     </head>
17
18     <body>
19         <h1><?php echo $this->placeholder('title') ?></h1>
20         <?php echo $this->layout()->content ?>
21
22         <br />
23         <br />
24     </body>
25
26 </html>
```

Zend_Layout

- **Two-step view pattern**
 - Uses Zend_View for rendering
- **Placeholders useful for setting javascript, titles, other variable data**
- **Layout view helper**
 - shortcut to layout placeholder
 - These are equivalent:

```
// fetch 'content' key using layout helper:  
echo $this->layout()->content;  
  
// fetch 'content' key using placeholder helper:  
echo $this->placeholder('Zend_Layout')->content;
```

Model

Model

- **Models are abstract representations of data**
 - Can be extended from:
 - Zend_Db_Table_Row – For database abstraction
 - Zend_Feed_Element – For RSS abstraction
 - Or any other class that fits your needs
 - Or build your own own abstract representations of your data
- **Model classes can contain business logic to prepare complex data for presentation**
- **I stuff any “weird” code in models so that controllers/views are clean**

Model: example

```
// model: Busyflag.php

class Busyflag
{
    protected $name = 'SYSFLAGS'; // old-fashioned "System 36"
    table

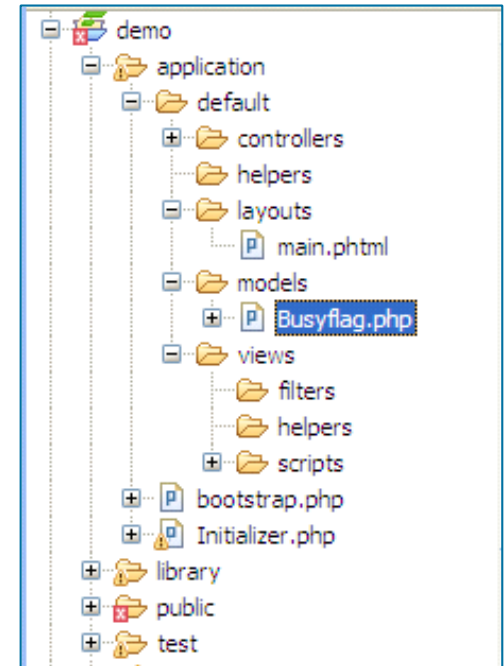
    // isSiteUp: return true if up, false if down
    public function isSiteUp() {

        $sql = "select BZYFLG from {$this->name} where RECID ='B'";
        $row = SBSDbhelp::getOneRow($sql);

        // true if Y, false otherwise.
        return $row['BZYFLG'] == 'Y';

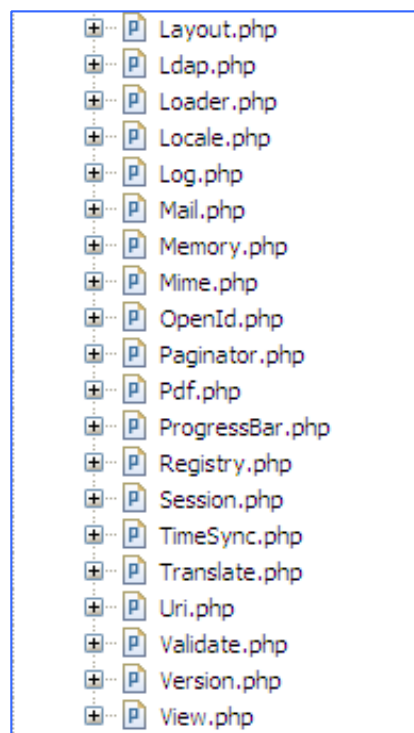
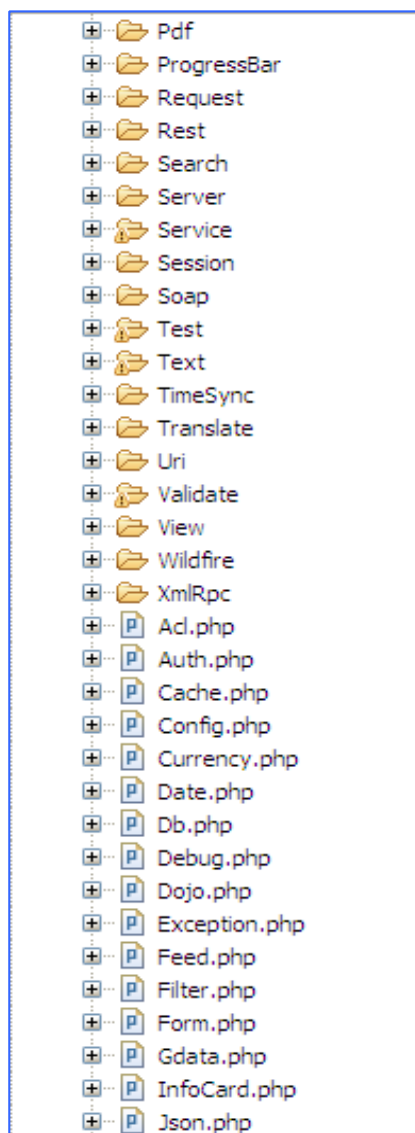
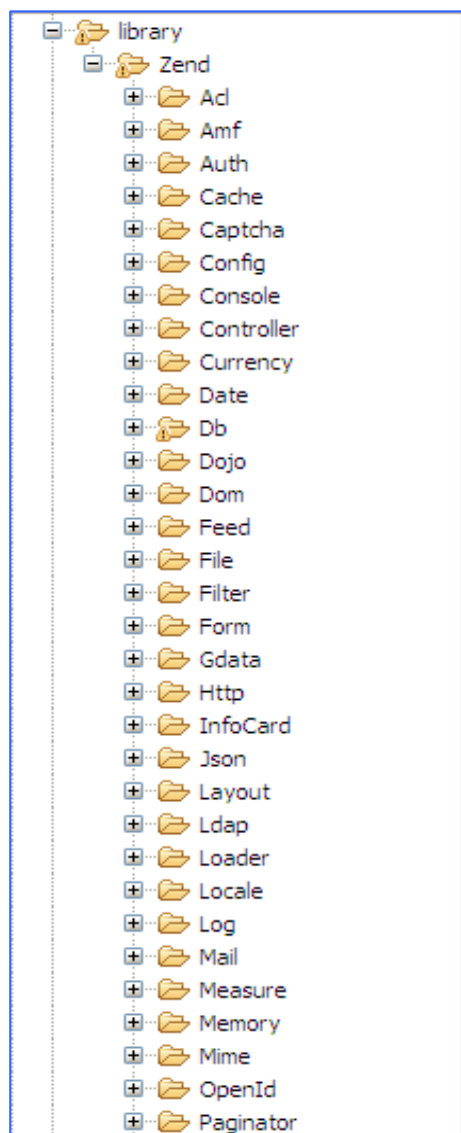
    } //(public function isSiteUp())
} //(class Busyflag)
```

```
// usage (from a preDispatch front controller plugin)
$busyFlag = new Busyflag();
if (!$busyFlag->isSiteUp()) {
    // Take user to "site down" page.
} //(if (!$busyFlag->isSiteUp()))
```



Components

Library of Zend components



Reminder:

Zend/Db.php = Zend_Db

Zend/Db/Table.php = Zend_Db_Table

Forms

Zend_Form

- **Creates the HTML for your data entry forms**
 - `$form = new Zend_Form();`
 - `$form->addElement('text', 'ordernum');`
 - `$form->addElement('text', 'date');`
- **Several ways to output form elements**
 - `echo $form; // (all elements) or`
 - `echo $form->ordernum; // (just ordernum) or`
 - `echo $form->getElement('ordernum');`
- **The HTML generated by that last echo**
 - `<input type="text" name="ordernum" id="ordernum">`

More complex Zend_Form example in MVC

```
// in a model:
class My_Form extends Zend_Form
{
    /* Create a text box that checks for non-letter characters
    ** and converts text to lower case on submission */
    $form->addElement('text', 'username', array(
        'validators' => array(
            'alnum',
            array('regex', false, '/^[a-z]/i')
        ),
        'required' => true,
        'filters' => array('StringToLower'),
    ));
}


// in a controller:
$form = new My_Form();
$this->view = $form

// in a view:
echo $this->form;
```

Real life example of Zend_Form

Search Product Catalog - Mozilla Firefox

Home Advanced Search Orders ABG Employee Connect Log out

 eBiz@ABG

Welcome, Angel Wong #105

Home > Search Product Catalog PRICING MONTH: **June 2009**

Keyword Search

Search for item: [?]
with bottle price (\$): [?] to [?] (optional) [\(proof °\)](#)

Advanced Search

Code	<input type="text"/> [?]	Description	<input type="text"/> [?]	Size	<input type="text"/> [?]
Color	<input type="text"/> [?]	Vintage	<input type="text"/> [?]	Place of Origin	<input type="text"/> [?]
Brand	<input type="text"/> [?]	Varietal	<input type="text"/> [?]	Supplier	<input type="text"/> [?]
Producer	<input type="text"/> [?]	Class	<input type="text"/> [?]		<input type="text"/> [?]
Kosher	<input type="checkbox"/> [?]	Sustainable	<input type="checkbox"/> [?]		
Value-Added Pack	<input type="checkbox"/> [?]	with bottle price (\$):			

Partial-description combo boxes (Dojo) on advanced search

- Charbono / Cort
- Charbono / Corbeau / Charbonneau
- Chardonnay
- Chardonnay Blend
- Chardonnay Sauvignon Blanc Blend
- Chardonnay-Pinot Grigio Blend
- Chardonnay-Pinot Gris Blend
- Chardonnay-Semillon Blend


Customer Service North: (800) 272-1323 South: (800) 841-1948
© Copyright Allied Beverage Group LLC. All Rights Reserved.

Search results

Search Results (14 products found) - Mozilla Firefox

http://ebizdev.alliedbeverage.com/advancedsearch/result/variatal/chardonni

Home Advanced Search Orders ABG Employee Connect Log out

 eBiz@ABG

Welcome, Angel Wong #105

Home > Search Product Catalog > Search Results (14 products found) PRICING MONTH: **June 2009**

Advanced Search

Search for item: [?]
 with bottle price (\$): [?] to [?] (optional) Search

Results for **Varietal: CHARDONNAY BLEND**

Product	Code	Size	Pack	Price		Inventory		Vintage	Info	\$ Best Buy
				\$ Case	\$ Bottle	Case	Bottle			
Antinori Cervaro Della Sala Chardonnay	6600543	750 MI	6			0	0	2006		
This product may be substituted with 6600544 .										
Bodegas Julian Chivite Navarra Reserva Coleccion 125	6784340	750 MI	6			5	3	NV		
Ca Montini Luna Di Luna Chardonnay/Pinot Grigio Gift Pack	4582449	750 MI	6			27	0	NV		1 cs: 149.99 149.99

Implementation of Product Id field

```
// AdvancedSearchForm class is a model:
class AdvancedSearchForm extends Zend_Form {

    $prodId = new Zend_Form_Element_Text("prodid",
        array('size' => 7, 'maxlength' => 7, 'class' =>
'width5'));

    $prodId->setRequired(false)
        ->addFilters(array("StripTags", "StringTrim"))
        ->addValidator(new Zend_Validate_Digits())
        ->setDescription("Partial product ID")
        ->setLabel("Code");

    $this->addElements(array($prodId));

} // (AdvancedSearchForm)
```

Database access

Database access with Zend_Db

- **Zend_Db can create SQL for you. You don't have to be an SQL expert to do everyday tasks**
- **Zend_Db offers a lot beyond creating SQL**
 - Consistent quoting, escaping, prepared statements, profiler
- **Eventually, you should try to become proficient in SQL, both to understand what Zend_Db is doing, and for creating more complex queries.**

- **Several classes give you a good start**
 - **Zend_Db_Adapter_Abstract**
 - Abstract class for all adapters
 - You will most likely use this or concrete implementations (such as Zend_Db_Adapter_Db2) for your database access
 - **Zend_Db_Table**
 - Gateway class for doing queries on a given table
 - **Zend_Db_Table_Row**
 - An instance of a given row
 - **Zend_Db_Statement**

Zend_Db_Table

- **Zend_Db_Table gives you record-level access similar to what you may be used to.**
 - **Insert**
 - **`$products->insert(array(`
`'prodid' => '1234567',`
`'prodname' => 'sparkling water',`
`);`**
 - **Update**
 - **Find (like chaining with a key)**
 - **`$results = $products->find('1234567');`**
 - **Delete**

More Zend_Db examples for i5

```
$driverOptions = array('i5_lib' => 'MYLIBRARY');  
// Use 'driver_options' => array('i5_naming' => DB2_I5_NAMING_ON)) for liblists  
  
$config = array(  
    'host'      => 'localhost',  
    'username' => 'ALAN',  
    'password' => 'secret',  
    'dbname'   => 'SBSDB',  
    'driver_options' => $driverOptions);  
  
$db = Zend_Db::factory('DB2', $config);  
  
// Using "select" method to select and display records  
$rows = $db->select()->from('CUSTOMERS')  
        ->where('CUSTNO >= 0');  
  
// or write your own SQL with parameters  
$sql = 'SELECT * FROM CUSTOMERS WHERE CUSTNO > ? and CUSTNO < ?';  
$rows = $db->fetchAll($sql, array(100, 2000));  
  
// either way, output results  
foreach ($rows as $row) {  
    echo $row['CUSTNO'] . ' ' . $row['CUSTNAME'];  
}
```

Config.ini lets you externalize Zend_Db settings

```
; config.ini
[dev]
db.adapter = PDO_MYSQL
db.params.username = alan
db.params.password = secret
db.params.dbname = devdb
db.params.host = 12.13.14.15

// in index.php (bootstrap file)
$config = new
    Zend_Config_Ini(realpath(dirname(__FILE__) .
        '/../application/config.ini'), 'dev');
$db = Zend_Db::factory($config->db);
```

Working with RPG


Use models to call RPG from ZF

- **I always wrap RPG calls in a model class to simplify my code. Here's why:**
 - If the RPG program's name changes, or we call a different program (e.g. CL instead of RPG), I only need to change the model class, not every place it's used
 - Implement consistent error handling (e.g. level check)
 - The model bridges the worlds of RPG and PHP
 - From PHP to RPG, zero-pad numbers
 - From RPG to PHP (return), interpret the RPG's results
 - Convert 'Y' to 'true'. Boolean values are well understood by PHP, can be evaluated by `if ($flag) ...`

Example of calling RPG from ZF

Build Order - Mozilla Firefox

Home Advanced Search Orders ABG Employee Connect Log out

 eBiz@ABG

Welcome, Angel Wong #105

Home > Orders > Build Order PRICING MONTH: **June 2009**

AMERICO'S INC (001588) Off Premise Terms: NET

DBA EUROPA LIQUORS
155-57 PACIFIC ST, NEWARK NJ, 07102
Phone: 973-589-8195
Messages: (none)

Ship From: NORTH Delivery Date: (days: TWHF) Mode: Comments (optional):

Add a product: cases Or select from: Search: History:

Dlt	Qty	Cs/Bt	Code	Msg	Description	\$ List	\$ Disc	\$ Net	\$ Ext	Inf	les Rep/House
✗	1	cases	3817220		JOSE CUERVO SILVER (1.75L/6)	249.99	28.00	182.99	182.99	\$	Angel Wong/30
✗	3	cases	5607061		A BY ACACIA CHD0712P (375ML/12)	67.99	00	67.99	197.97	\$	Angel Wong/30
Add 1 or more cases to qualify for the 2 case \$20 RIP.											
*NOTE: Product is a half case qualifier.											
✗	1	bottles	7433065		CAKEBREAD CHR07 12P (375ML/12)	16.99	00	16.99	16.99	\$	105: Angel Wong/30

Customer Service North: (800) 272-1323 South: (800) 841-1948
© Copyright Allied Beverage Group LLC. All Rights Reserved.

Model hides the details of calling RPG

```
class wer104
{
    public function __construct($sequence = 0) {

        // lots of code in here, conversions, error handling, etc.
        . . . $parmsIn = array('PWEBID'=>$sessionKey);
        . . . I5_command . . .

        $this->_isValidData = (($returnValues['PRTN'] == 'Y') ? true
: false);
        // be very explicit, true or false

    }

    final public function isValidData()
    {
        return $this->_isValidData;
    }

} //(class wer104)
```

See how simple the controller code is

```
/* in controller, use model 'wer104'
 * which wraps/calls RPG */
$validationCall = new wer104($sequence);

if (!$validationCall->isValidData()) {
    // validation failed; redirect to "edit"
    . . .
}

// otherwise, we passed validation...
```

Paginator

Zend_Paginator

- **Handles page-at-time logic, similar to subfiles, for large lists**
- **Gives you:**
 - the right data records
 - Page numbering, back, next, first, last
- **For data, it's commonly “fed” an array or db select object**
 - If database select, paginator is smart enough to read only the records to be displayed on the page

Example of Zend_Paginator code

Controller

```
$result = $db->select()->from("SLEMSTP");

$paginator = Zend_Paginator::factory($result);

// Set parameters for paginator
$paginator->setCurrentPageNumber($this->getParam("page")); // URL must be something like:
    http://example.com/orders/index/page/1  <- meaning we are currently on page one, and pass that
    value into the "setCurrentPageNumber"
$paginator->setItemCountPerPage(20);
$paginator->setPageRange(10);

// Make paginator available in views
$this->view->paginator = $paginator;
```

View script

```
<?php if (count($this->paginator)): ?>
<ul>
<?php foreach ($this->paginator as $item): ?>
    <li><?= $item['LENAME1']; ?></li>
<?php endforeach; ?>
</ul>
<?php endif; ?>

<?= $this->paginationControl($this->paginator, 'Sliding', 'partials/paginationcontrol.phtml'); ?>
```

Example of Zend_Paginator code

View Partial (used in View Script on previous slide)

NOTE these view helpers: `$this->url` which build URL links with the prev, next, and other page numbers, and leads back to controller with the page clicked by user.

```
<?php echo sprintf('Page %s of %s', $this->current, 'xxx'); ?>
<?php if ($this->pageCount): ?>
<div class="paginationControl">
<!-- Previous page link -->
<?php if (isset($this->previous)): ?>
    <a href="<?=$this->url(array('page' => $this->previous)); ?>">&lt; Previous</a> |
<?php else: ?>
    <span class="disabled">&lt; Previous</span> |
<?php endif; ?>

<!-- Numbered page links -->
<?php foreach ($this->pagesInRange as $page): ?>
    <?php if ($page != $this->current): ?>
        <a href="<?=$this->url(array('page' => $page)); ?>"><?=$page; ?></a> |
    <?php else: ?>
        <?=$page; ?> |
    <?php endif; ?>
<?php endforeach; ?>

<!-- Next page link -->
<?php if (isset($this->next)): ?>
    <a href="<?=$this->url(array('page' => $this->next)); ?>">Next &gt;</a>
<?php else: ?>
    <span class="disabled">Next &gt;</span>
<?php endif; ?>
</div>
<?php endif; ?>
```

Zend_Paginator display

(The appearance can be fully customized by changing the View and View Partial scripts)

- [Piccolo Tuesday Men's "D"](#)
- [Ramapo Valley White Sox](#)
- [Huntsville International League](#)
- [Streetsboro Flames](#)
- [Fontana Community Little League](#)
- [Test](#)
- [The Bandits](#)
- [Black Hat](#)
- [Brookline Mens Softball](#)
- [Intensity](#)
- [Lou Gehrig League](#)
- [Mel Ott League](#)
- [Roy Campanella League \(AAA\)](#)
- [Roy Campanella League \(A\)](#)
- [patch](#)
- [Spring League 2001](#)
- [Lehigh Valley MSBL](#)
- [Lancaster Depew Leagues](#)
- [Roy Campanella](#)
- [Spartans](#)

url:

<http://example.com/leagues/index/page/5>

[< Previous](#) | [1](#) | [2](#) | [3](#) | [4](#) | [5](#) | [6](#) | [7](#) | [8](#) | [9](#) | [10](#) | [Next >](#)



**Other components
you'll like**

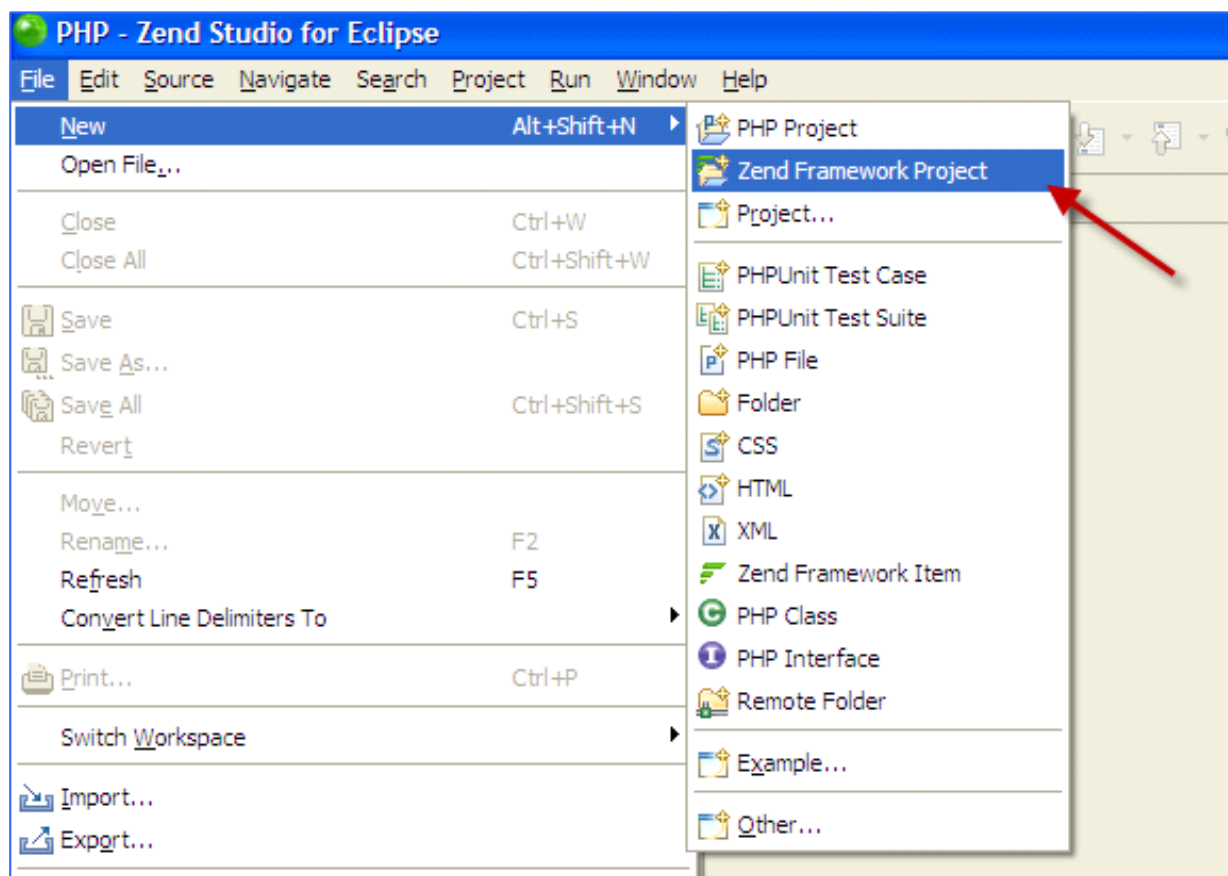
Other components

- **Auth**
- **ACL**
- **Filter/Validate**
- **Log (with familiar concept of logging levels)**
- **Navigation (bread crumbs)**

How to start a ZF project?

Start the right way with Zend Studio for Eclipse

- **Creates a complete “hello world” application for you**
 - Leverage the ZF development team’s best practices



Resources: online

- **Official information:**
 - framework.zend.com/docs/quickstart
 - zend.com/resources/webinars
- **Community tutorials and answers:**
 - zfforums.com
 - devzone.zend.com

Path to ZF

- **Jump in**
 - Have a pilot project in mind
 - Take a ZF training class
 - Get mentoring from someone savvy in both ZF and “i”
- **Stay connected**
 - Join a ZF community, either online or a Meetup in person
 - Subscribe to Zend’s ZF support if it’s a mission-critical app
 - Contact Mike and me for guidance: mike.p@zend.com, aseiden@sbsusa.com

Questions and Thanks



Mike: mike.p@zend.com

Alan: aseiden@sbsusa.com

Leave a comment: alanseiden.com/presentations