



# Zend Framework for Enterprise PHP on IBM i



By Alan Seiden - PHP/i Developer and Consultant

Strategic Business Systems, Inc.

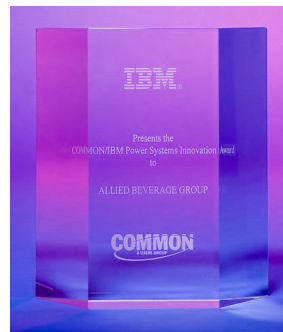


ZEND/PHP CONFERENCE OCTOBER 15-22, 2011

# About me (Alan Seiden)

---

Consultant/developer and mentor specializing in  
PHP & Zend Framework on IBM i



- Consultant to IBM's Redbook: *Zend Core for i5/OS*
- First IBM i developer certified in Zend Framework
- Co-founder of first Zend Framework Meetup (New York)
- Developed "best web app" (COMMON meeting 2009)

# About Strategic Business Systems, Inc.

---

## Partner with IBM since 1982

- ▶ IBM i hardware, software development, consulting
- ▶ Concentration in food & beverage and automotive industries
- ▶ HQ in northern New Jersey

## Partner with Zend (“the PHP company”) since 2008

- ▶ PHP’s been our preferred web technology since 2005
- ▶ In addition to our consulting/development services, we help customers select and purchase Zend’s training and software
- ▶ We represent Zend in the northeastern USA

# What I'll cover today

---

## Evolution of PHP on IBM i

- ▶ In general, and...
- ▶ For each developer, assess where you are and what's next

## Zend Framework

- ▶ Why it's good for you
- ▶ Concepts and components
- ▶ Tips and tricks

**You will leave here understanding what ZF can do—and that you can do it!**

# Why PHP on i

---

Powerful, flexible, popular

Runs completely on your good old “i”

Has backing & support of IBM

Allows your teams to work together

- ▶ Re-use RPG logic, called from PHP
- ▶ Consolidate departmental solutions (Excel spreadsheets!) into your reliable db2 database

# Typical stages of PHP on i

---

## Grass roots

- ▶ Prove to yourself how well PHP works on i

## Pilot project(s)

- ▶ Obtain some buy-in from management

## Enterprise PHP

- ▶ Important project developed with experience

# Grass roots stage

---

# Grass roots stage

---

## Go ahead, kick the tires!

- ▶ Download and install Zend Core (free)
- ▶ Create a simple web page reading/writing to db2
- ▶ Try open source software (such as Mantis/400)
- ▶ Prove it to yourself before “selling” to manager

## Special tools are optional at this point

- ▶ Even Notepad will do, but Zend Studio is free on IBM i

# Grass roots resources

---

**Zend Core:** <http://zend.com/products/core/downloads>

- ▶ One year of free “silver” support included

## Online help and information

- ▶ PHP Manual: <http://php.net>
- ▶ *Zend Core for i5/OS* Redbook
- ▶ Knowledge base: <http://kb.zend.com>
- ▶ Forums (look for i5 forums): <http://forums.zend.com>

# Pilot Project

---

# Pilot project: how to pick one

---

Well defined, useful, with modest scope

Accesses IBM i resources

- ▶ db2
- ▶ RPG calls (optional)

Could be a new application or extension of an existing one to the web

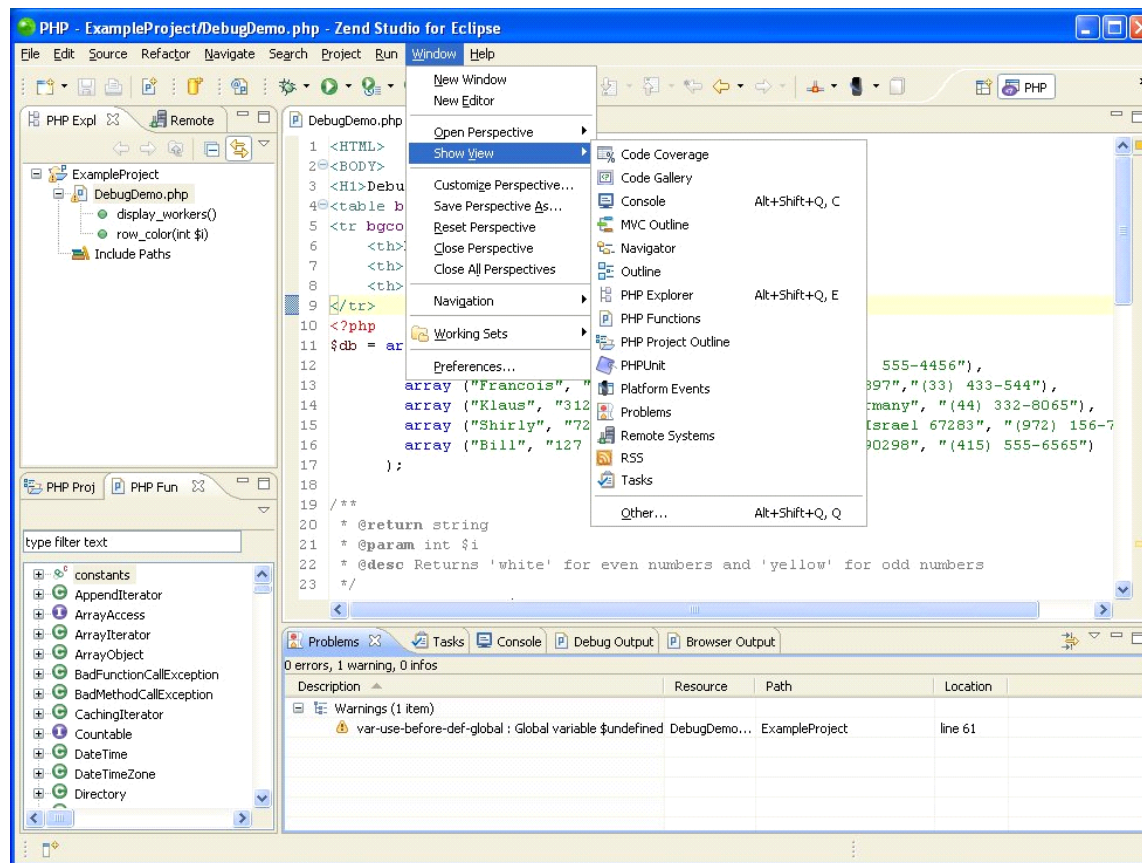
# Pilot project: goals and resources

---

- **Small scale lets you do it right**
  - ▶ Set standards for future
  - ▶ Safe space to get experience
- **You need these free tools**
  - ▶ Zend Core, Zend Studio
- **And one or more of these:**
  - ▶ Training/Support
  - ▶ A PHP framework (e.g. Zend Framework, which is free)
  - ▶ Mentoring, coached development
    - Greatly accelerate progress, save staff time
    - Especially when starting with a framework
  - ▶ Zend Platform (try 60-day trial)

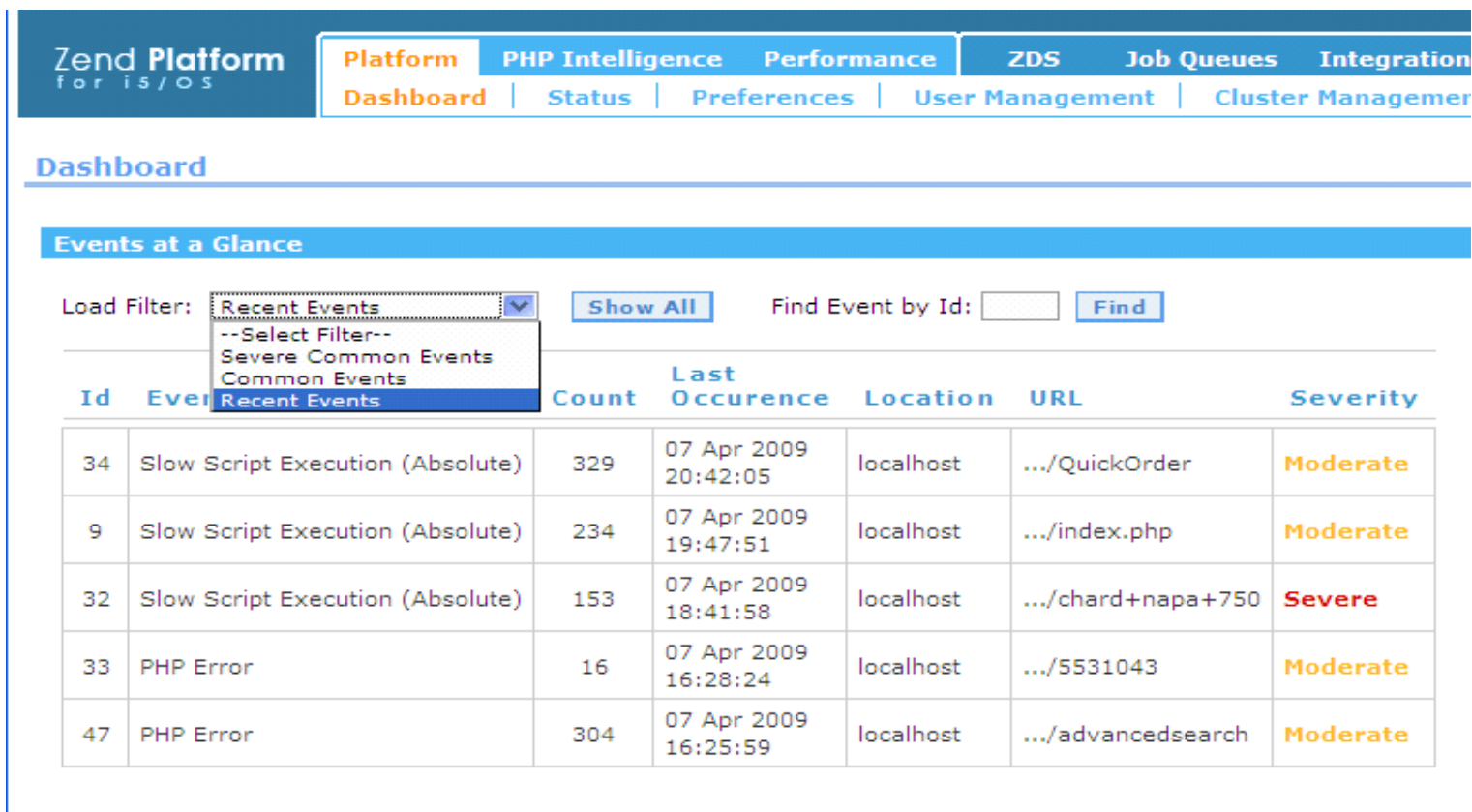
# Zend Studio for Eclipse (IDE)

- Highlights errors, offers interactive debugging, and speeds creation of Zend Framework projects



# Zend Platform (soon to be Zend Server)

Provides caching (fast!), debugging and event tracking, integration with Zend Studio (IDE)



The screenshot displays the Zend Platform dashboard for i5/OS. The navigation menu includes 'Platform', 'PHP Intelligence', 'Performance', 'ZDS', 'Job Queues', and 'Integration'. The 'Dashboard' section is active, showing 'Events at a Glance'. A dropdown menu for 'Load Filter' is open, listing 'Recent Events', '--Select Filter--', 'Severe Common Events', 'Common Events', and 'Recent Events'. Below the filter is a 'Show All' button and a search field for 'Find Event by Id:'. The main content is a table of events with columns for Id, Event, Count, Last Occurrence, Location, URL, and Severity.

Id	Event	Count	Last Occurrence	Location	URL	Severity
34	Slow Script Execution (Absolute)	329	07 Apr 2009 20:42:05	localhost	.../QuickOrder	Moderate
9	Slow Script Execution (Absolute)	234	07 Apr 2009 19:47:51	localhost	.../index.php	Moderate
32	Slow Script Execution (Absolute)	153	07 Apr 2009 18:41:58	localhost	.../chard+napa+750	Severe
33	PHP Error	16	07 Apr 2009 16:28:24	localhost	.../5531043	Moderate
47	PHP Error	304	07 Apr 2009 16:25:59	localhost	.../advancedsearch	Moderate

# Zend Framework

---

- A starting point for your applications, providing
  - ▶ Security features
  - ▶ Modular design
- 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

But what is Enterprise PHP?

---

# Enterprise PHP is:

---

- Grown-up development
- Think of your mature RPG processes
  - ▶ Consistent
  - ▶ Reliable
- You know how to do that

# Enterprise PHP in detail

---

## Design/development cycle

- ▶ Standards
- ▶ Security
- ▶ Focus on easy maintainance
- ▶ Reusable functionality (not just copy and paste)
- ▶ Procedures that ensure quality

## Deployment

- ▶ Need tools to help support app in live environment
- ▶ Predictable, consistent performance

# Zend Framework

---

An important piece of the enterprise puzzle

# The PHP community is evolving

---

Old question: “what is PHP?”

Current question: “Which PHP Framework do you use?”

New question: “How do I integrate “x” framework into a project with Zend Framework?”

# Three benefits of a PHP framework

---

## 1. Modular design

- ▶ Avoid the tangled mess that comes with growth

## 2. Best practices

- ▶ You absorb the practices and (we hope) good habits of the framework's creators.

## 3. Components

- ▶ Pre-built functionality
- ▶ Keep up with new technology for you

# Where to get it

---

- Pre-installed with Zend Core
  - ▶ `/usr/local/Zend/ZendFramework/library`
  - ▶ `php.ini` includes it in `include_path`
- Get the latest at [framework.zend.com](http://framework.zend.com)
  - ▶ Copy into `/usr/local/Zend/ZendFramework/library` or install elsewhere
  - ▶ Can install in your application's library folder as well
  - ▶ Make sure you set include path properly

# 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)

# Zend Framework 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
  - ▶ Oct. 2009: version 1.94; minor releases every couple of weeks
- **April 2009: ZF/i application won COMMON's "best web solution"**

# COMMON award for best web solution 2009

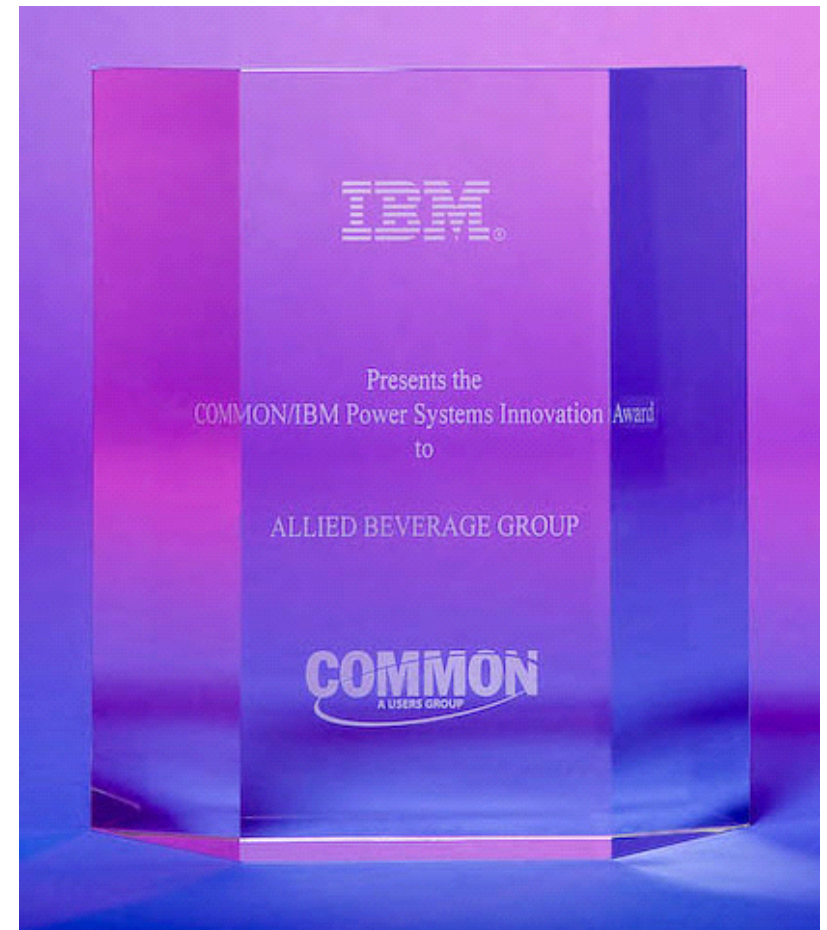
---

Allied Beverage Group's  
"eBiz"

Wine catalog/ordering system  
on IBM i

Built in Zend Framework

Full-text catalog search in  
native db2

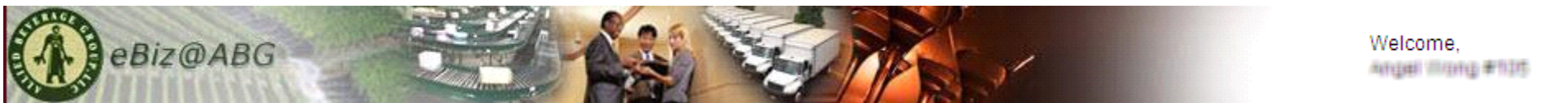


# Why eBiz was successful

---

- For Allied, PHP brings people and resources together on the “i”
  - ▶ db2 database provides reliable, consistent product catalog and order data
  - ▶ No extra servers needed
  - ▶ Re-using RPG logic
  - ▶ Everyone on the same team
  - ▶ PHP is “superglue”: plans afoot to connect other systems, too
- In the wine distribution industry, a solid reputation is important
  - ▶ For their first high-profile project, asked Strategic (us) to develop the system
- Consistency, standards, security, reusability
  - ▶ Zend Framework
- Speed critical for sales reps in the field
  - ▶ (Being compared to green screen speed!)
  - ▶ Zend Platform

# Allied Beverage eBiz live demo (if time)



Welcome,  
Angel Wong #135

Home > Search Product Catalog > Search Results (7 products found), order for EUROPA LIQUORS (001588)

PRICING MONTH: **June 2009**

## Advanced Search

Search for item:  <sup>[?]</sup> [\(proof<sup>o</sup>\)](#)  
 with bottle price  <sup>[?]</sup> to  <sup>[?]</sup> (optional)

**AMERICO'S INC (001588)**  
 DBA EUROPA LIQUORS  
 155-57 PACIFIC ST, NEWARK NJ, 07102  
 Phone: 973-686-4145

Off Premise Terms: NET  
 Total List: \$215.54  
 Total Disc: \$25.00  
**Est Total Net: \$ 190.54**  
 [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			
<a href="#">Acacia Chardonnay A By Acacia</a>	5607061	375 MI	12	<input type="text" value="3"/>	cases ▾	<input type="button" value="Add"/>			26	5	NV		
<a href="#">Cakebread Cellars Chardonnay Napa Valley 07</a>	7433065	375 MI	12	<input type="text" value="1"/>	bottles ▾	<input type="button" value="Add"/>			9	5	2007		
<a href="#">Graich Hills Cellar Chardonnay</a>	5544265	375 MI	12	<input type="text"/>	cases ▾	<input type="button" value="Add"/>			5	7	2006		
<a href="#">Levendi Chardonnay Red Hen 05</a>	4591060	375 MI	12	<input type="text"/>	cases ▾	<input type="button" value="Add"/>			13	2	2005		
<a href="#">Merrvale Chardonnay Starmont 07</a>	4223069	375 MI	12	<input type="text"/>	cases ▾	<input type="button" value="Add"/>			8	8	2007		
<a href="#">Schramsberg Blanc de Blanc</a>	4056064	375 MI	12	<input type="text"/>	cases ▾	<input type="button" value="Add"/>			17	9	2005		



# Object orientation (OO)

---

Instant intro: only two 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.

# Model-View-Controller

---

Good structure for your application

# Model - View - Controller (MVC) design pattern

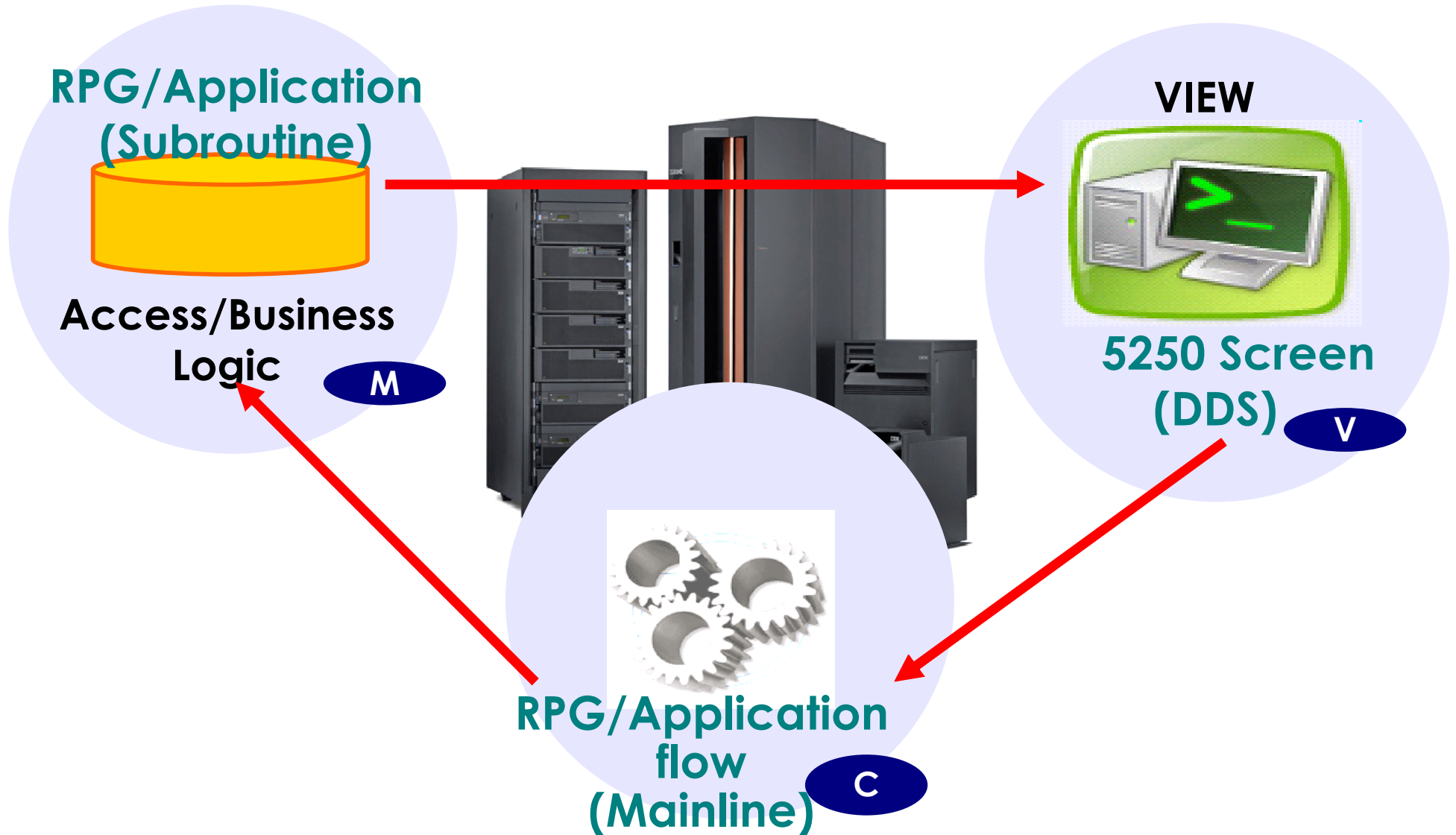
- You already know this pattern from RPG/DDS
- With green screens, IBM handles it under the covers, so you take it for granted
- On the web, you must define your application's structure more explicitly
- Be patient...MVC seems strange at first, but you'll soon realize that you've been here before...

# MVC in detail

---

- **Model**
  - ▶ Reusable classes that access data and 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

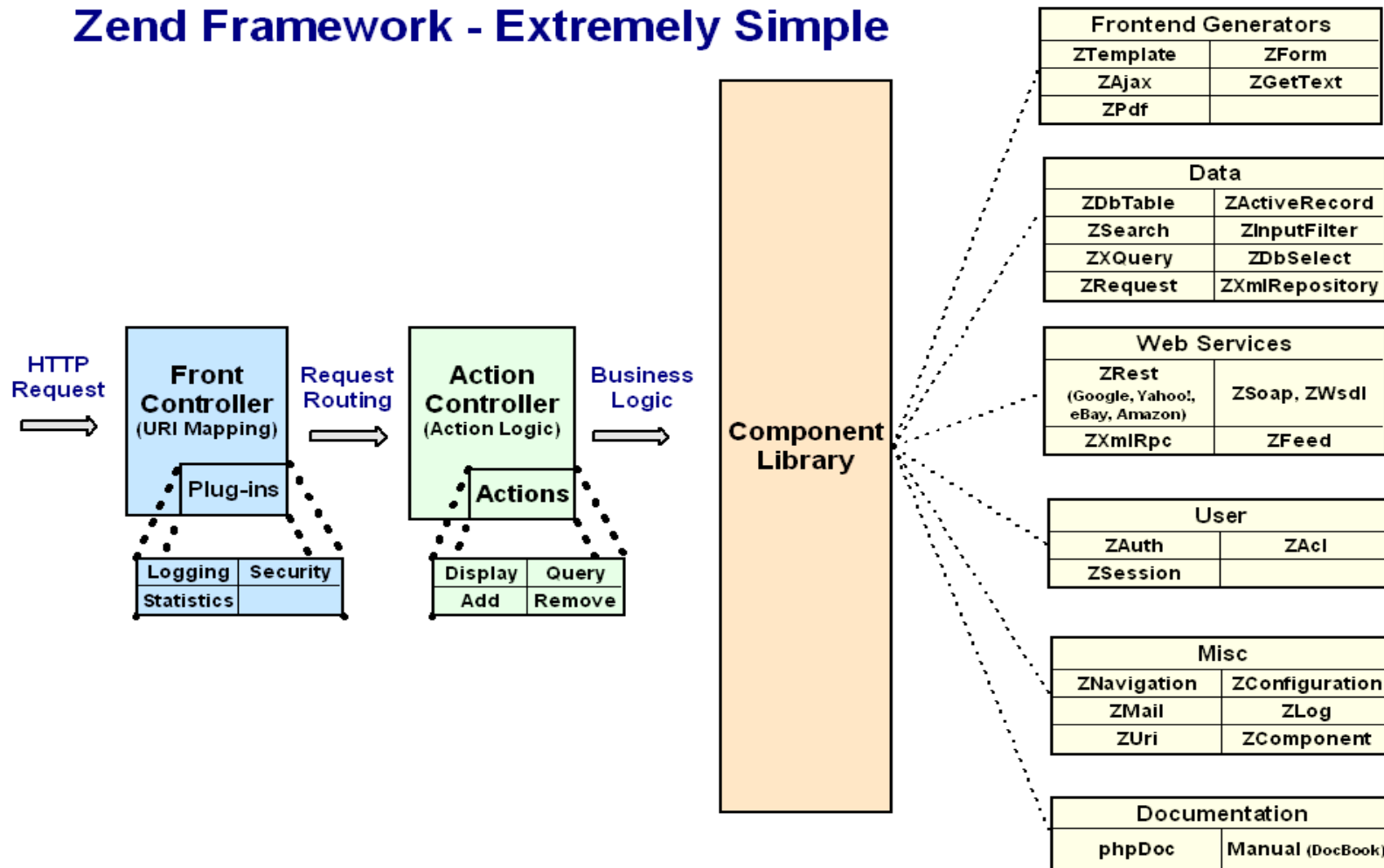
- ▶ Simplified the controller, which was getting complex and hard to understand
- ▶ Made the SQL reusable

# Initialize MVC

---

# Front controller to action controller

## Zend Framework - Extremely Simple



# Front controller routes “friendly” URL

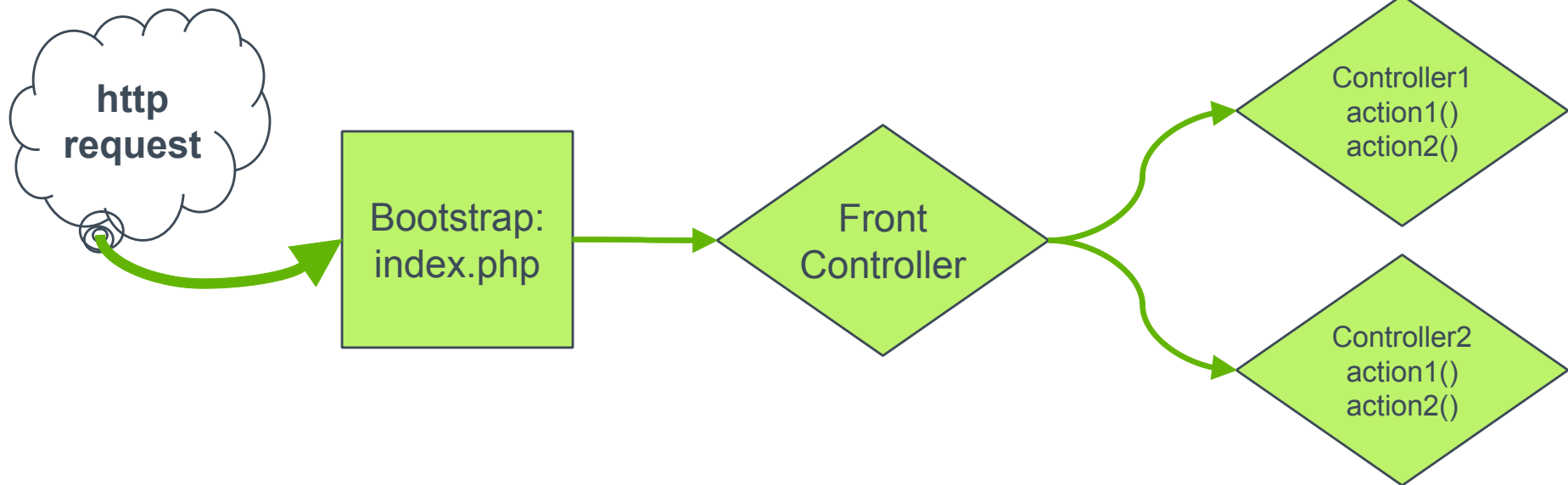
- Default routing convention:

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

Controller maps  
to class name

Action maps to  
method name

Param/value pairs  
are passed to  
action

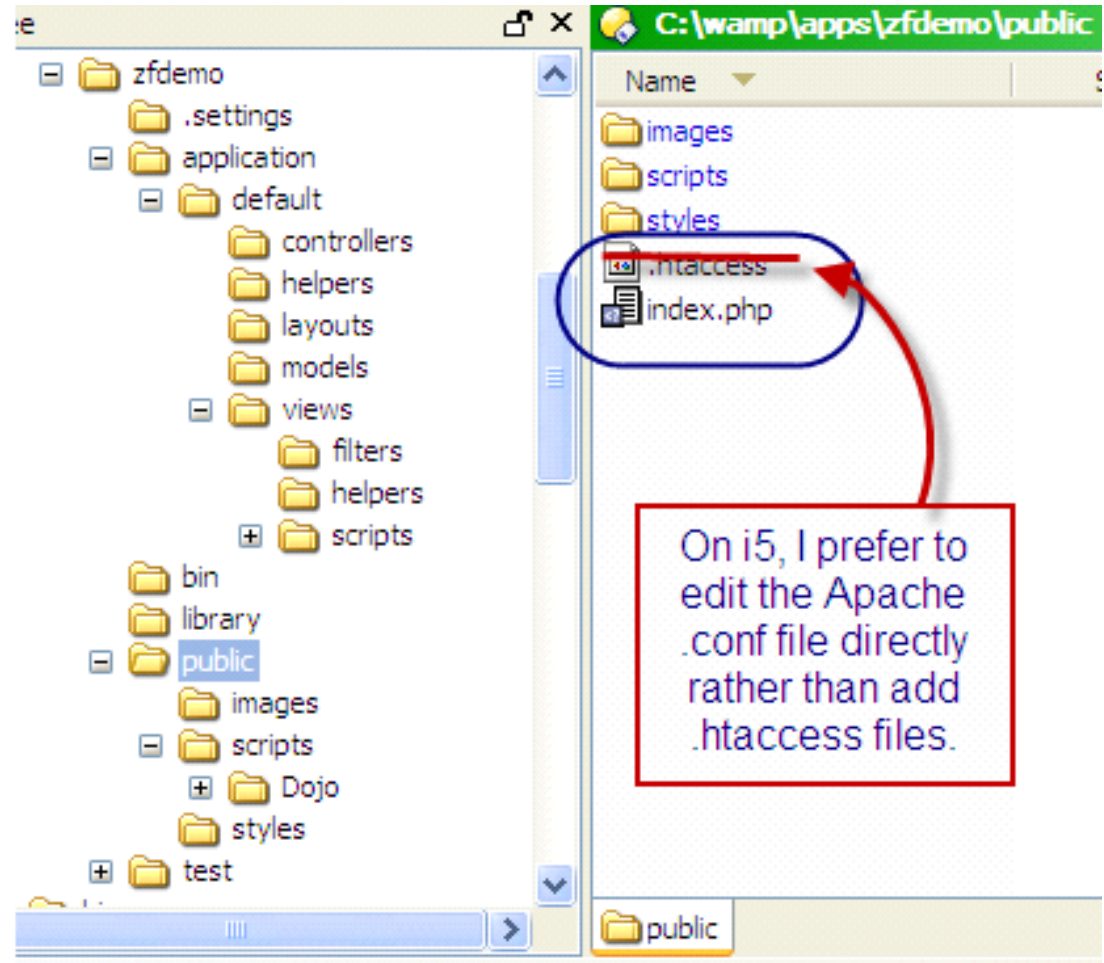


# All requests through index.php in doc root

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)
// use explicit, full paths to 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

---

Zend\_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

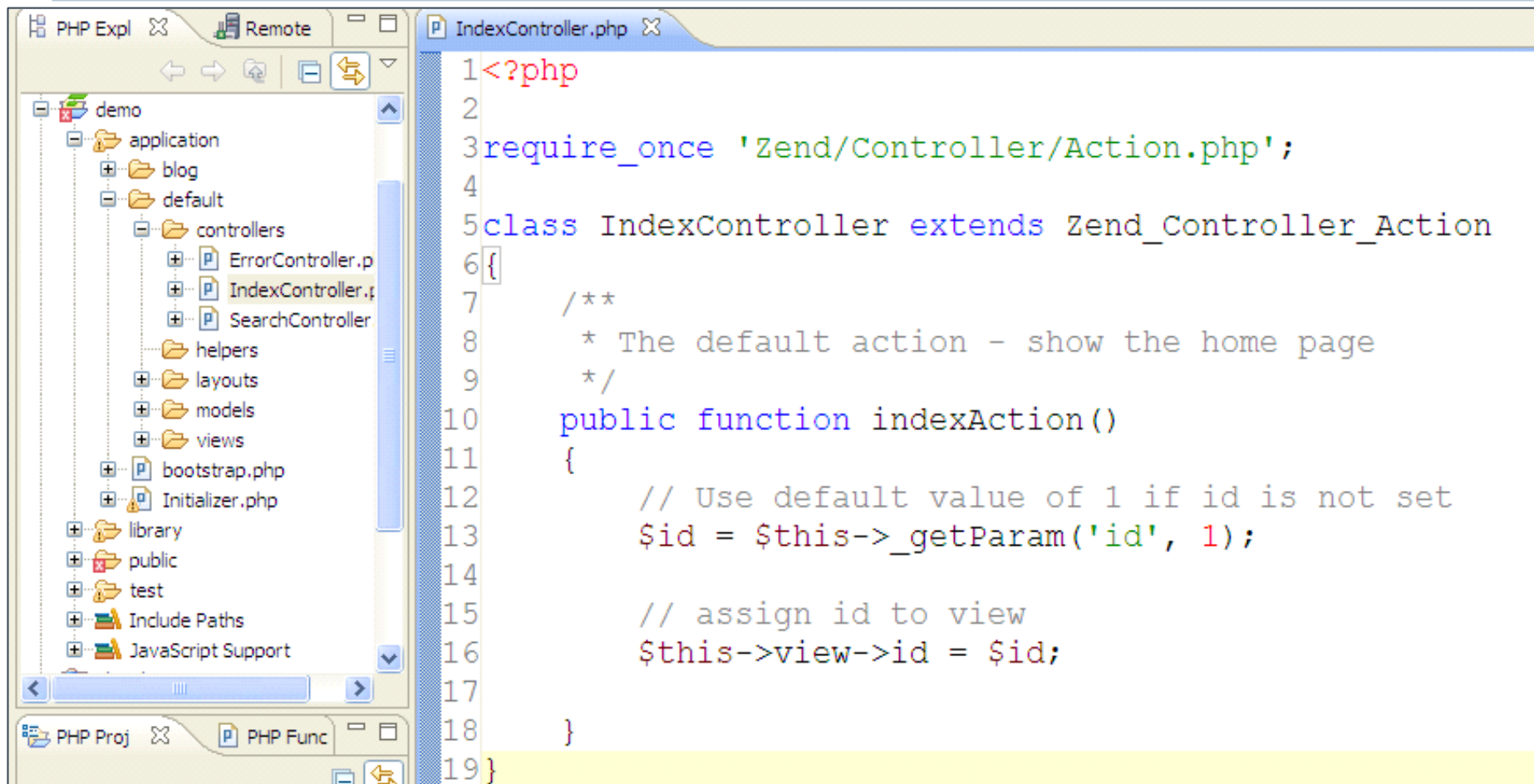
- **Action methods in each controller class handle requests**

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

Default: `indexAction()`

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

# Controller example



The screenshot shows the PHP Explorer interface with a project named 'demo'. The project structure includes folders for 'application', 'blog', 'default', 'helpers', 'layouts', 'models', 'views', 'library', 'public', and 'test'. The 'default' folder contains a 'controllers' sub-folder with files 'ErrorController.p', 'IndexController.p', and 'SearchController'. The 'IndexController.php' file is open in the editor, showing the following code:

```
1<?php
2
3require_once 'Zend/Controller/Action.php';
4
5class 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

---

Zend\_View

# 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

## 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.

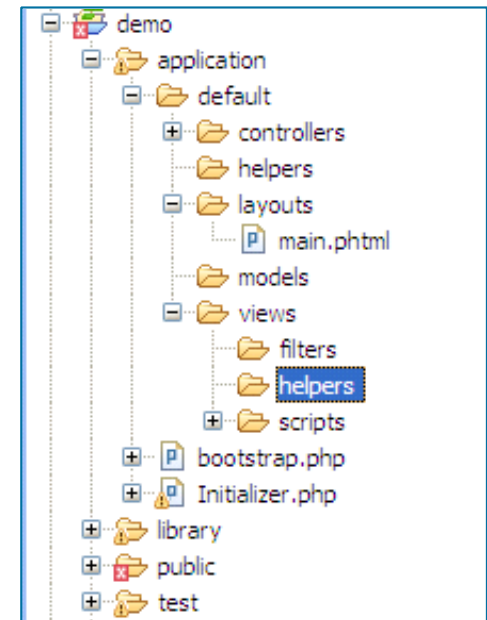
# My own view helper: TitleCase.php

```
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;
    }
}
```

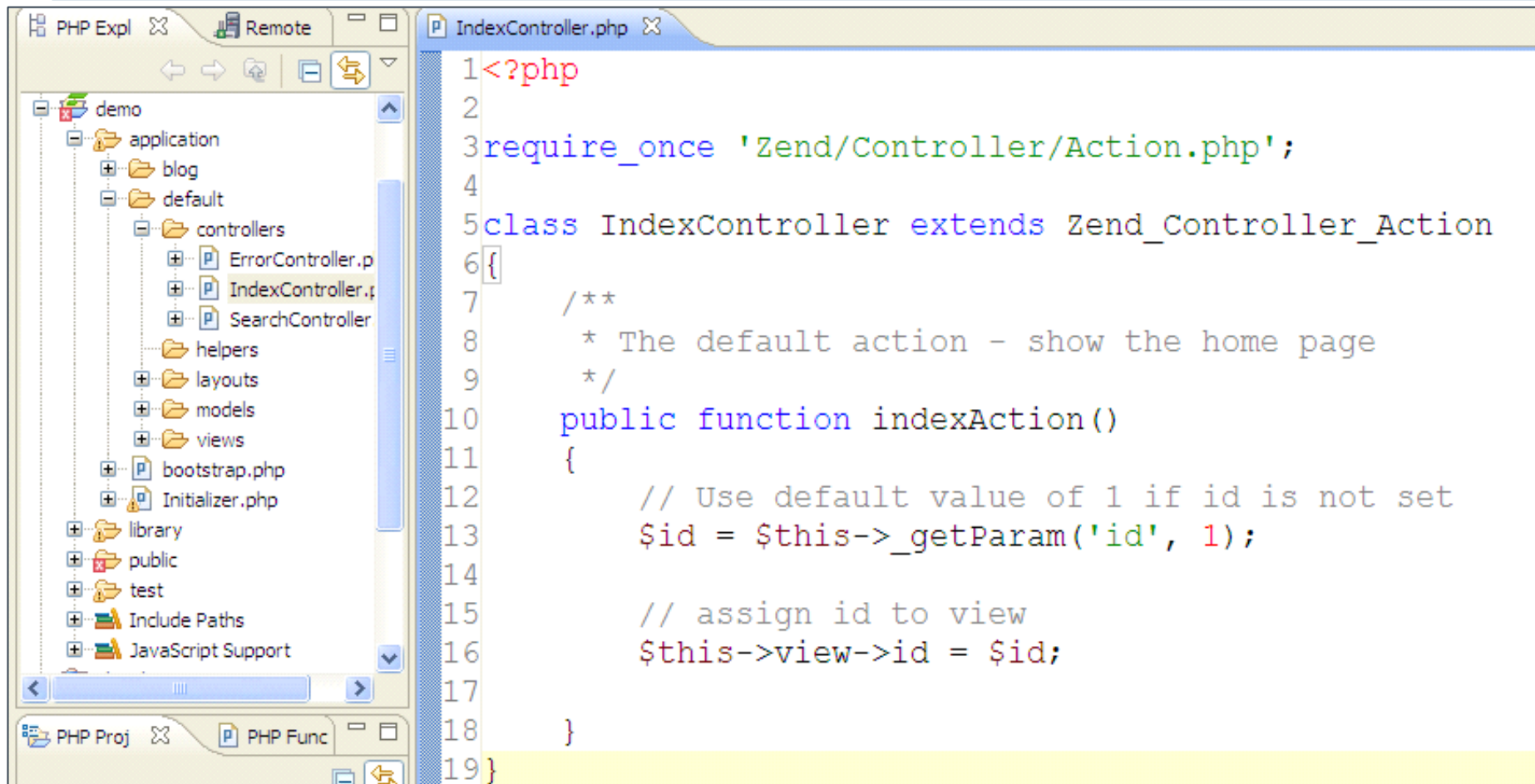


## Usage:

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

// Mozilla Firefox
```

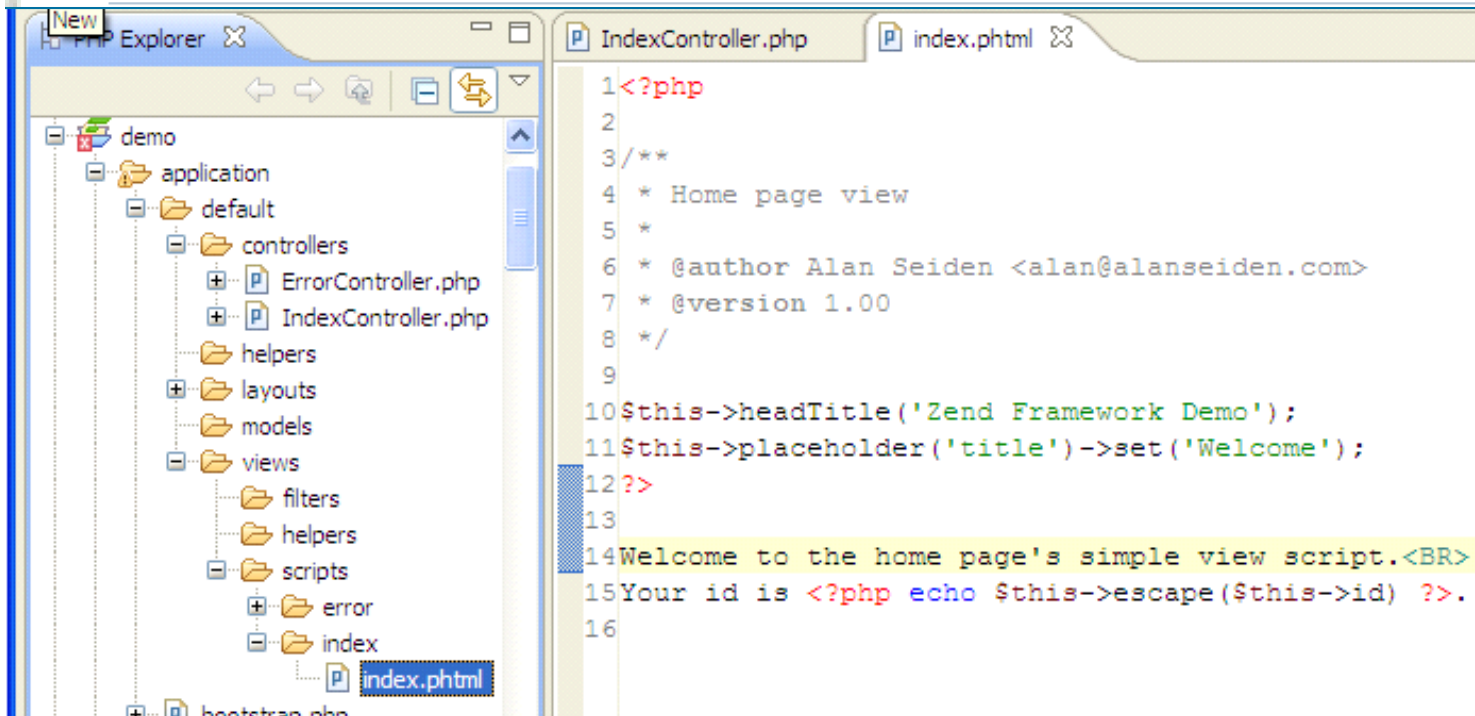
# Controller (again)...leads to view



The screenshot shows the PHP Explorer interface. On the left, a tree view displays the project structure for a 'demo' application. The 'application' folder is expanded to show 'default', which contains 'controllers'. Under 'controllers', three files are listed: 'ErrorController.p', 'IndexController.p', and 'SearchController'. The 'IndexController.p' file is selected. The main editor window on the right shows the code for 'IndexController.php'. The code is as follows:

```
1<?php
2
3require_once 'Zend/Controller/Action.php';
4
5class 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



```
1 <?php
2
3 /**
4  * Home page view
5  *
6  * @author Alan Seiden <alan@alanseiden.com>
7  * @version 1.00
8  */
9
10 $this->headTitle('Zend Framework Demo');
11 $this->placeholder('title')->set('Welcome');
12 ?>
13
14 Welcome to the home page's simple view script.<BR>
15 Your id is <?php echo $this->escape($this->id) ?>.
16
```



Zend Framework Demo	Zend Framework Demo
<p>http://localhost/zfdemo/</p> <p><b>Welcome</b></p> <p>Welcome the home page's simple view script. Your id is 1.</p>	<p>http://localhost/zfdemo/index/index/id/39</p> <p><b>Welcome</b></p> <p>Welcome to the home page's simple view script. Your id is 39.</p>

# Zend\_Layout

---

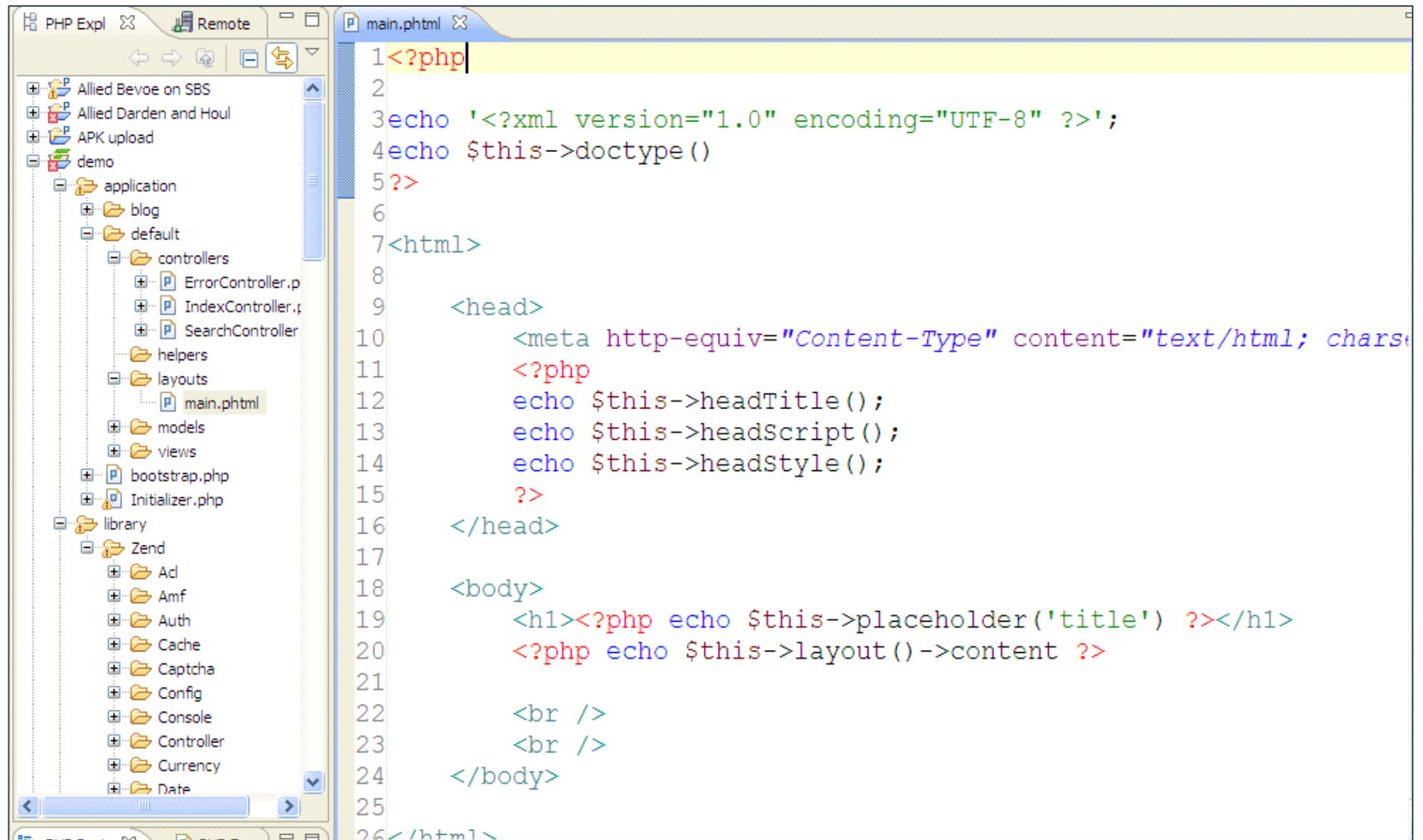
Gives your site a consistent look while retaining flexibility

## Two-step view pattern

- ▶ Uses Zend\_View for rendering

Placeholders useful for setting javascript, titles, other variable data

# Zend\_Layout code



The screenshot shows the PHP Explorer interface with a file tree on the left and a code editor on the right. The file tree shows a project structure with folders like 'application', 'library', and 'Zend'. The code editor displays the content of 'main.phtml'.

```
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>
```

# Model

---

# Model

---

**Model classes are abstract representations of data**

- ▶ Can be extended from:
  - Zend\_Db\_Table - For database 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: “system busy” flag

```
// model: Busyflag.php

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

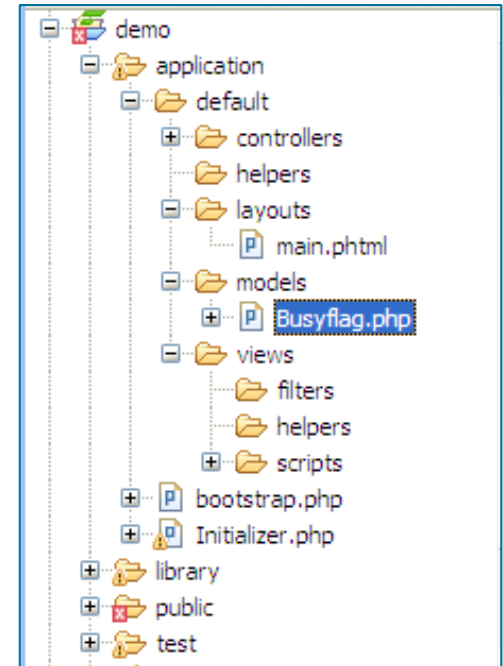
    // isSiteUp: return true if up, false if down
    public function isSiteUp() {

        $sql = "select BZYFLG from {$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()))
```

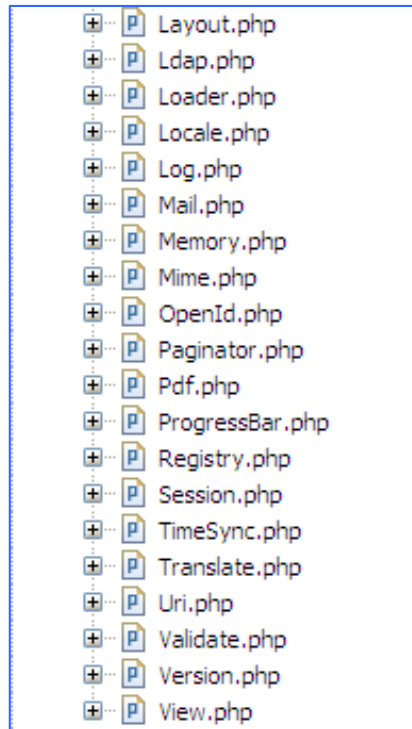
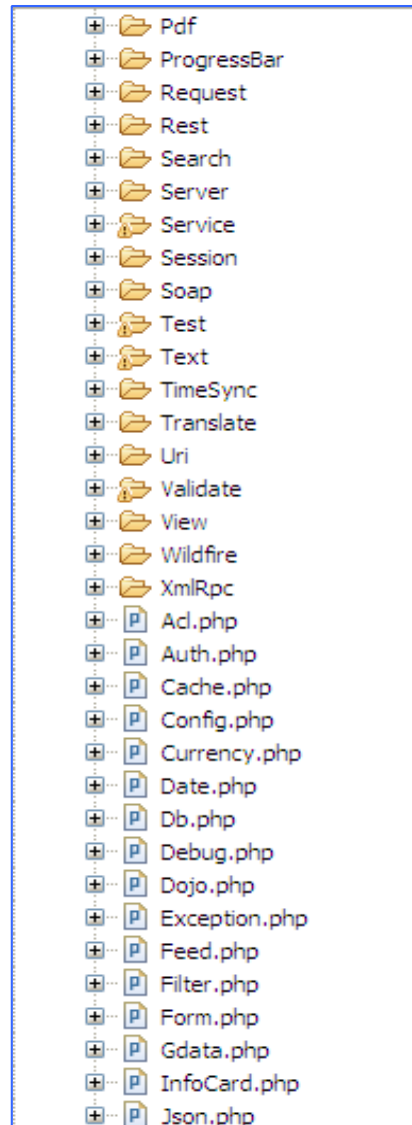
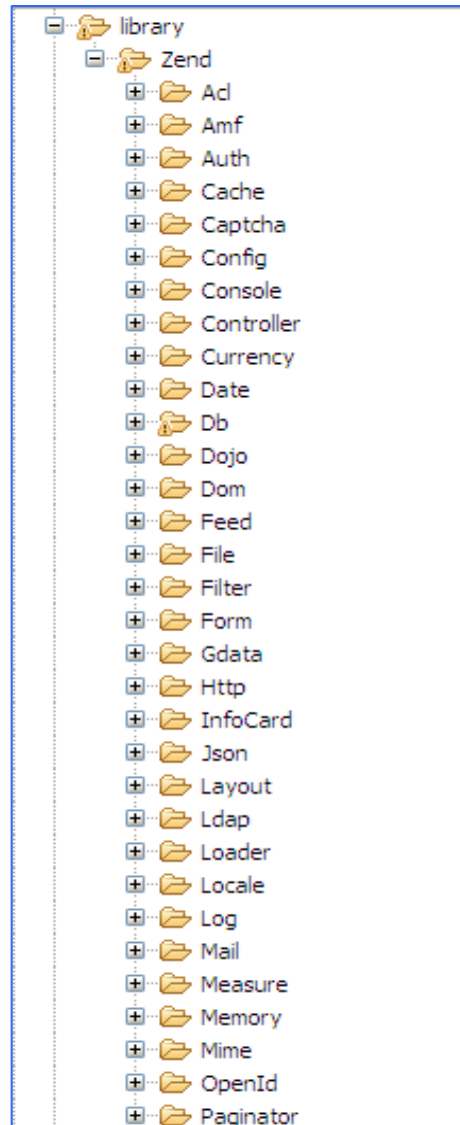


# Components included in ZF

---

Dozens so far...with more on the way

# Library of Zend components



**Reminder:**

**Zend/Db.php = Zend\_Db**

**Zend/Db/Table.php = Zend\_Db\_Table**

# Forms

---

Zend\_Form component

# 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

The screenshot shows a web browser window titled "Search Product Catalog - Mozilla Firefox". The page has a navigation bar with links for Home, Advanced Search, Orders, ABG Employee Connect, and Log out. Below the navigation bar is a banner image with the eBiz@ABG logo and a welcome message for Angel Wong #105. The main content area features a "Keyword Search" section with a search input field and a "Search" button. Below this is an "Advanced Search" section with various filters: Code, Color, Brand, Producer, Kosher, Value-Added Pack, Description, Vintage, Varietal, Class, Sustainable, Size, Place of Origin, Supplier, and with bottle price (\$). A dropdown menu for "Varietal" is open, showing a list of wine varieties including Charbono / Corbeau / Charbonneau, Chardonnay, Chardonnay Blend, Chardonnay Sauvignon Blanc Blend, Chardonnay-Pinot Grigio Blend, Chardonnay-Pinot Gris Blend, and Chardonnay-Semillon Blend. A callout box points to the dropdown menu with the text "Partial-description combo boxes (Dojo) on advanced search". At the bottom of the page, there is contact information for Customer Service North and South, and a copyright notice for Allied Beverage Group LLC.

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:  [?] (proof\*)

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

**Advanced Search**

Code  [?]

Color  [?]

Brand  [?]

Producer  [?]

Kosher  [?]

Value-Added Pack  [?]

Description  [?]

Vintage  [?]

Varietal  [?]

Class  [?]

Sustainable  [?]

with bottle price (\$):  [?]

Size  [?]

Place of Origin  [?]

Supplier  [?]

[?]

**Partial-description combo boxes (Dojo) on advanced search**

- 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/chardon... snagit blur

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:  [?] (proof °)

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

Results for Varietal: CHARDONNAY BLEND

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

---

Zend\_Db component

# Database access with Zend\_Db

---

Zend\_Db can create SQL for you

- ▶ Fewer long, error-prone SQL strings

Zend\_Db offers a lot beyond creating SQL

- ▶ consistent quoting
- ▶ escaping
- ▶ prepared statements (substitutions for “?”)
- ▶ profiler

# Database access

---

## Several classes give you a good start

- ▶ `Zend_Db_Adapter_Abstract`: uniform API for many RDBMS's
  - `Zend_Db_Adapter_Db2`
  - `Zend_Db_Adapter_Pdo_Mysql`
- ▶ `Zend_Db_Table`
  - Gateway class for doing queries on a given table
- ▶ `Zend_Db_Statement`
  - Specify SQL for “prepared” statement (secure--with ? marks)
- ▶ ...and more

# 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

# Zend\_Db techniques for IBM i

---

```
$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 externalizes 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

---

Wrap i5 Toolkit calls in model classes

# 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

 Welcome, Angel Wong #105

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

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

DBA EUROPA LIQUORS    Total List:   
 155-57 PACIFIC ST, NEWARK NJ, 07102    Total Disc:   
 Phone: 973-589-8195    Est Total Net \$   
 Messages: (none)    Cases: 4 Bottles: 1

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

Add a product:  cases     Or select from:     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)					\$	105: Angel Wong/30
✗	3	cases	5607061		A BY ACACIA CHD0712P (375ML/12)					\$	105: 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)						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, we use model 'wer104'
 * which wraps/calls RPG */
$validationCall = new wer104($sequence);

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

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

# Pagination

---

Zend\_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

# Zend\_Paginator display

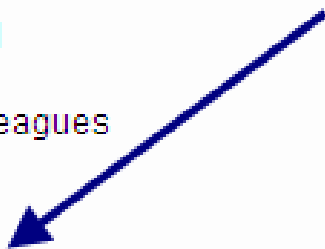
(The appearance can be fully customized by changing the View and View Partial scripts)

- [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 >](#)



# 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'); ?>
```

# Code to build links for Paginator

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; ?>
```

# Other components you'll like

---

# Zend\_Registry

---

- Store global data without polluting the global scope
  - ▶ `Zend_Registry::set('key', $value);`
  - ▶ `Zend_Registry::get('key');`
  - ▶ Useful for storing sessions, configuration data or any data that could potentially be important on a per-request basis

# Zend\_Log

---

- A structured, flexible logging class with several backends
  - ▶ Stream (write to a file)
  - ▶ Firebug
  - ▶ Mail
  - ▶ Db
- Example of logging to file system

```
$writer = new Zend_Log_Writer_Stream('/path/to/logfile');  
// Only send entries of warning level and higher.  
$writer->addFilter(Zend_Log::WARN);  
$logger = new Zend_Log($writer);  
  
// later...  
$logger->info('Informational message');  
$logger->err('Error message');  
  
// "Error message"
```

# Zend\_Config

---

- OO syntax for reading/writing config files
- Internally it's an array
- Multiple backends: INI, XML
- Divide sections with [ ], good for dev/prod
  - ▶ And hierarchy with periods after that
    - `db.params.username = xxxx`
- Many components accept Zend\_Config object as well as array
  - ▶ Db
  - ▶ Forms

# Config + Registry

---

Config.ini:

```
[dev]
```

```
db.adapter = PDO_MYSQL
```

```
db.params.username = admin
```

```
db.params.password = xxxx
```

```
db.params.dbname = appdb
```

```
db.params.host = 192.168.9.1
```

```
log.filename = /appdev/logs/app.log
```

```
$config = new Zend_Config_Ini(realpath(dirname(__FILE__) . '/../application/config.ini'),  
    'dev');
```

```
$registry = Zend_Registry::getInstance();
```

```
$registry->set('config', $config);
```

```
$db = Zend_Db::factory($config->db);
```

```
Zend_Db_Table::setDefaultAdapter($db);
```

```
$registry->set('db', $db);
```

```
// create logger; store it in registry
```

```
$logFile = $registry->get('config')->log->filename;
```

```
$writer = new Zend_Log_Writer_Stream($logFile);
```

```
$logger = new Zend_Log($writer);
```

```
$registry->set('logger', $logger);
```

# Caching with Zend\_Cache

---

## Frontend

- ▶ Core
- ▶ Output
- ▶ Function
- ▶ File
- ▶ Class

## Backend

- ▶ Apc
- ▶ File
- ▶ Memcached
- ▶ Sqlite
- ▶ ZendPlatform/Server

# Zend\_Cache: basic code example

---

```
$frontendOptions = array(
    'lifetime' => 7200
);
$backendOptions = array(
    'cache_dir' => '../application/cache/'
);

$cache = Zend_Cache::factory('Output', 'File', $frontendOptions,
    $backendOptions);

if (!$cache->start('product list')) {
    // this content will be held till end() method called
    echo 'writing cacheable content...';

    $cache->end(); // save and output content
}
}
```

# Zend\_PDF

---

```
<?php
// combine three TIFF images into a multi-page PDF
$images = array('/images/page1.tif', '/images/page2.tif', '/images/page3.tif');

$newFileName = '/pdf/full_document.pdf';

$pdf = new Zend_Pdf();

foreach ($images as $image) {

    // Create new page
    $pdfPage = $pdf->newPage(Zend_Pdf_Page::SIZE_A4);
    $image = Zend_Pdf_Image::imageWithPath($image);
    $pdfPage->drawImage($image, 10, 10, 400, 500);

    // add page to pdf
    $pdf->pages[] = $pdfPage;

} //(foreach ($images as $image))

// Save document as a new file
$pdf->save($newFileName);
```

# Client classes for web services

---

- Akismet
- Amazon
- Audioscrobbler
- Delicious
- Flickr
- Google Data
- Nirvanix
- ReCaptcha
- Simpy
- SlideShare
- Strikelron
- Technorati
- Twitter
- Yahoo

# Zend\_Service\_Yahoo

---

## Search the web with Yahoo

- ▶ Get your application ID from <http://developer.yahoo.com/wsregapp/>
- ▶ Class uses Zend\_Rest\_Client under the covers
- ▶ Returns Zend\_Service\_Yahoo\_WebResultSet containing instances of Zend\_Service\_Yahoo\_WebResult

```
$yahoo = new Zend_Service_Yahoo("YAHOO_APPLICATION_ID");
$results = $yahoo->webSearch('IBM PHP',
                               array('results' => '10',
                                       'start'   => 1));

foreach ($results as $result) {
    echo '<b>' . $result->Title . '</b> ' . $result->Url . '<br />';
}
```

# Results from \$yahoo->webSearch

---

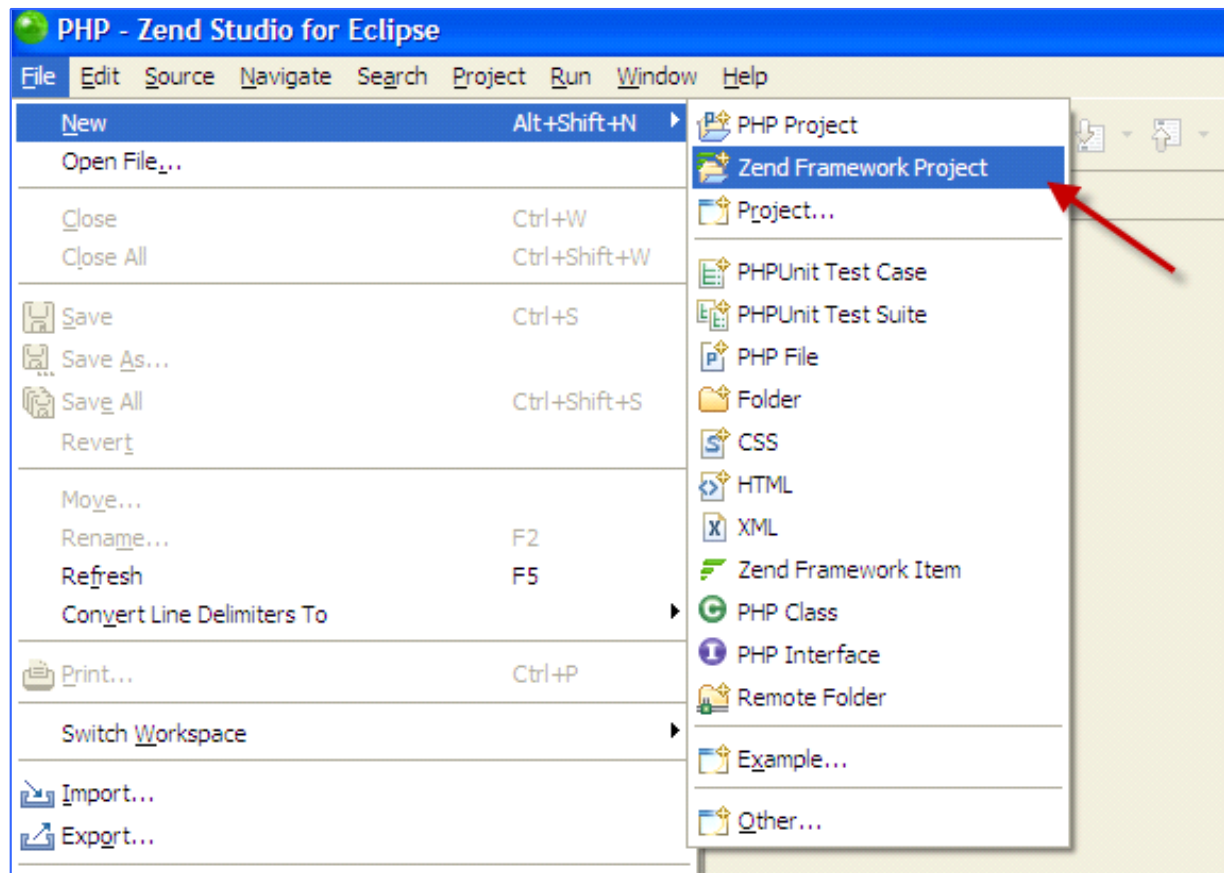
**IBM developerWorks : Blogs : Patrick Mueller** <http://www.ibm.com/developerworks/blogs/page/pmuellr?tag=php>  
**Informed Networker - Social News for IT Professionals. - IBM, Zend ...** <http://www.informednetworker.com/other/ibplatform/>  
**:: News : PHP : Zend Core for IBM on Linux** <http://madpenguin.org/cms/?m=show&id=4775>  
**:: News : PHP : IBM backs open-source Web software** <http://madpenguin.org/cms/index.php/?m=show&id=3567>  
**Digg - IBM: PHP development within Eclipse** [http://digg.com/programming/IBM\\_PHP\\_development\\_within\\_Eclipse](http://digg.com/programming/IBM_PHP_development_within_Eclipse)  
**Digg - Will IBM Buy Zend / PHP ?** [http://digg.com/linux\\_unix/Will\\_IBM\\_Buy\\_Zend\\_PHP](http://digg.com/linux_unix/Will_IBM_Buy_Zend_PHP)  
**php, simplexml | Diigo** <http://www.diigo.com/tag/php+simplexml>  
**ibm, rest | Diigo** <http://www.diigo.com/tag/ibm,rest>  
**IBM poop heads say LAMP users need to "grow up"** <http://naeblis.cx/rtomayko/2005/05/28/ibm-poop-heads>  
**Hypergene MediaBlog " IBM, blogging and the rise of the world's biggest ...** <http://www.hypergene.net/blog/print.ph>

# 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](http://framework.zend.com/docs/quickstart)
- ▶ [zend.com/resources/webinars](http://zend.com/resources/webinars)

## Community tutorials and answers:

- ▶ [zfforums.com](http://zfforums.com)
- ▶ [devzone.zend.com](http://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

# Questions?

---

Leave feedback at [joinin.com/900](https://www.joinin.com/join/900)

For free PHP/i tips, write to [phptips@alanseiden.com](mailto:phptips@alanseiden.com)



PHP/i blog and articles: [alanseiden.com](http://alanseiden.com)

Email: [alan@alanseiden.com](mailto:alan@alanseiden.com)

LinkedIn: [linkedin.com/in/alanseiden](https://www.linkedin.com/in/alanseiden)

SBS office phone: 1-800-727-7260