Why Use Zend Framework?

Alan Seiden

PHP on IBM i consultant/developer

email: aseiden@sbsusa.com

blog: http://alanseiden.com

Strategic Business Systems, Inc.

- **Developing Web apps on IBM i (and iSeries, i5...) since 1996**
- **Based in Ramsey, New Jersey**

- **IBM Business Partner**
    - Power Systems hardware, software, development, consulting

- **Zend Business Partner**
    - Working with Zend since they brought PHP to IBM i in 2006
    - We offer Zend's training and Zend Server software to complement our consulting/development services

Alan Seiden

PHP on IBM i Developer / Consultant / Mentor

- **Contributor to IBM's Redbook** *PHP: Zend Core for i5/OS*
- **First IBM i developer certified in Zend Framework**
- **Developer of IBM/COMMON's "Best Web Solution" of 2009**
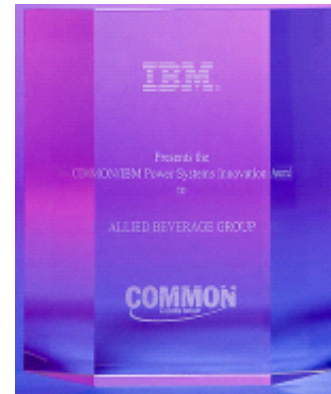
Contact: aseiden@sbsusa.com or 201-327-9400

Blog/articles: http://alanseiden.com

What we'll discuss today

- **Zend Framework as the premier PHP framework**

- **Why ZF is a great match for the IBM i**

- **What ZF can do for your PHP/i projects**

- **How to get started!**

- **Questions**

What I've been doing lately with PHP and ZF

Wine distributor transaction search

Displaying scanned, signed invoices

Authenticate to Windows AD

- **Options come from config.ini file:**

```
ldap.server1.host = ad.alliedbeverage.com
ldap.server1.useStartTls = false
ldap.server1.accountDomainName = mydomain.com
;…more
```

- **Authenticate to AD!**

```
$qualifiedAdUsername = 'MYDOMAIN\\username"
$authAdapter = new Zend_Auth_Adapter_Ldap($options,
    $qualifiedAdUsername, $password);
$auth=Zend_Auth::getInstance();
$result=$auth->authenticate($authAdapter);
```

Zend Framework

- **A free, open source, cross-platform collection of components written in PHP**
  - Over 70 PHP components to simplify common tasks, including:
    - Web form creation
    - Database access
    - Web services
    - Multi-language (English, French, Spanish…)

- **Included with Zend Server for IBM i and at http://framework.zend.com**

More than components

- **Optionally, a starting point for your applications, providing <u>modular design</u>**

- **It provides standards and functionality but does not limit you**

- **"Use at will" philosophy**

More benefits of ZF

- **Safe license for corporate use**
  - Use ZF's code however you like
  - http://framework.zend.com/license

- **High quality, unit tested**

- **It keeps up with trends and APIs**
    - Compatibility with diverse database systems, authentication standards (Windows AD) and APIs (e.g. Twitter)

Zend Framework Community

- **Contributors include individuals and companies. Companies include:**
  - Zend (of course)
  - IBM
  - OmniTI

- **Technology partners:**
  - Adobe, Google, IBM, Microsoft, nirvanix, StrikeIron

- **Individuals:**
  - Me! Improvements for IBM i developers

Here's why ZF reminds me of the "i" 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

- **Support available—from Zend**
  - Similar to phoning IBM about i/OS

Why ZF's time is right

- **PHP is being used for critical apps on IBM i**

- **It's time for professional practices**
  - Standards and consistency
  - Awareness of security
  - Reuse and easy maintenance of code
  - Training and support
  - Doing it "right"

- **ZF gets you there—"Enterprise PHP"—faster—and keeps you in control**

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

- **2007-2010: Fast progress**
  - July 2007: GA version 1.0
  - Feb. 2009: version 1.70 with db2/i5 support
  - Oct. 2010: version 1.11.0; 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
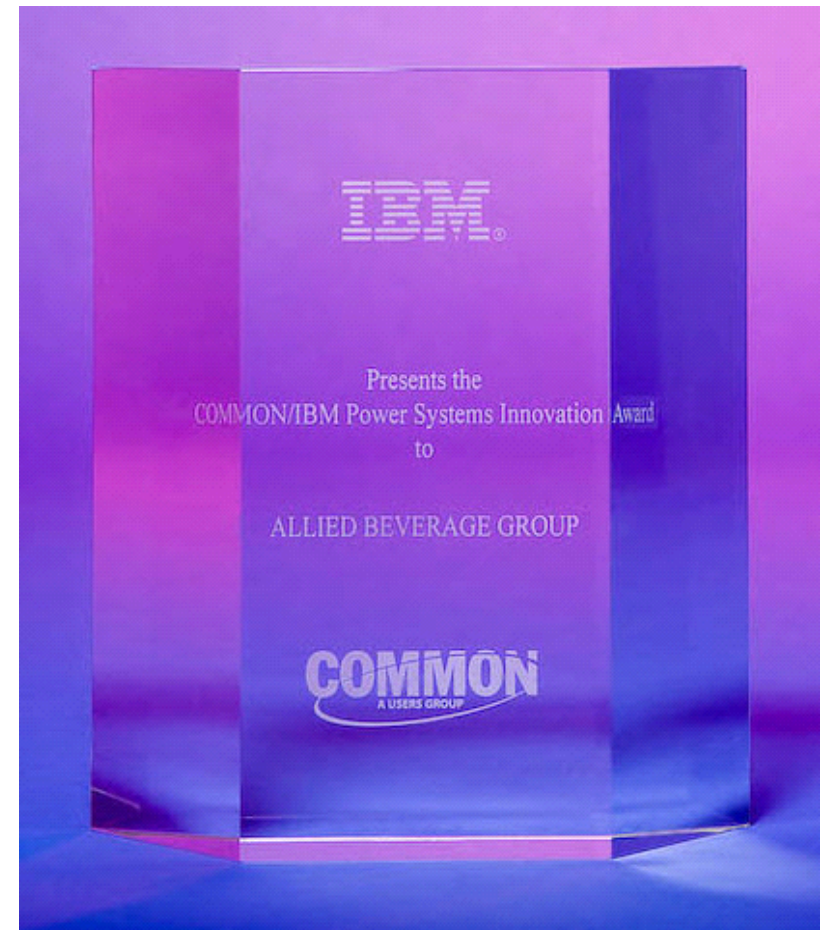
Allied Beverage eBiz live demo (if time)

Object Orientation
(Instant 2-slide intro)

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**	`$_orderNum`
Method	**function or subprocedure**	`isOrder()`

Class	**Imagine an intelligent data structure containing both data (properties) and programming logic (methods), which are both called "members" of the class**	`class Order {`
		`protected $_orderNum;`
		`function isOrder() {`
		`    . . .`
		`    }`
		`. . .`
		`}`

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 the notation above, you can read ZF code. You will learn to appreciate its simplicity.**

For more on OO PHP

See Mike Pavlak's "Intro to OO PHP" here at OMNI 1:15 pm

Convenient to include ZF

ZF is pre-loaded and ready to use

On IBM i, ZF comes pre-loaded with Zend Server in
`/usr/local/zendsvr/share/ZendFramework/library`

This "`library`" folder is pre-pathed in PHP.INI:

```
include_path =
    ".:/usr/local/zendsvr/share/ZendFramework/library
        :/usr/local/zendsvr/share/pear"
```

This path lets your PHP applications find all the ZF components.

Autoloader handles "require/include" for you

- **Normally you have to require or include any PHP classes you use.**
  - E.g. `require_once 'Zend/Form.php';`
  - Then `$form = new Zend_Form;`

- **Autoloader avoids this**
  - Follow PEAR convention for class/file names
  - E.g. Zend_Form means Zend/Form.php
  - Put this in bootstrap file ONCE:

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

  - Now you won't need a require or include statement to do:

```
$form = new Zend_Form(); // works for your own classes too!
```

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)

RPG/Application
(Subroutine)

Access/Business
Logic **M**

VIEW

5250 Screen
(DDS) **V**

RPG/Application
flow
(Mainline) **C**
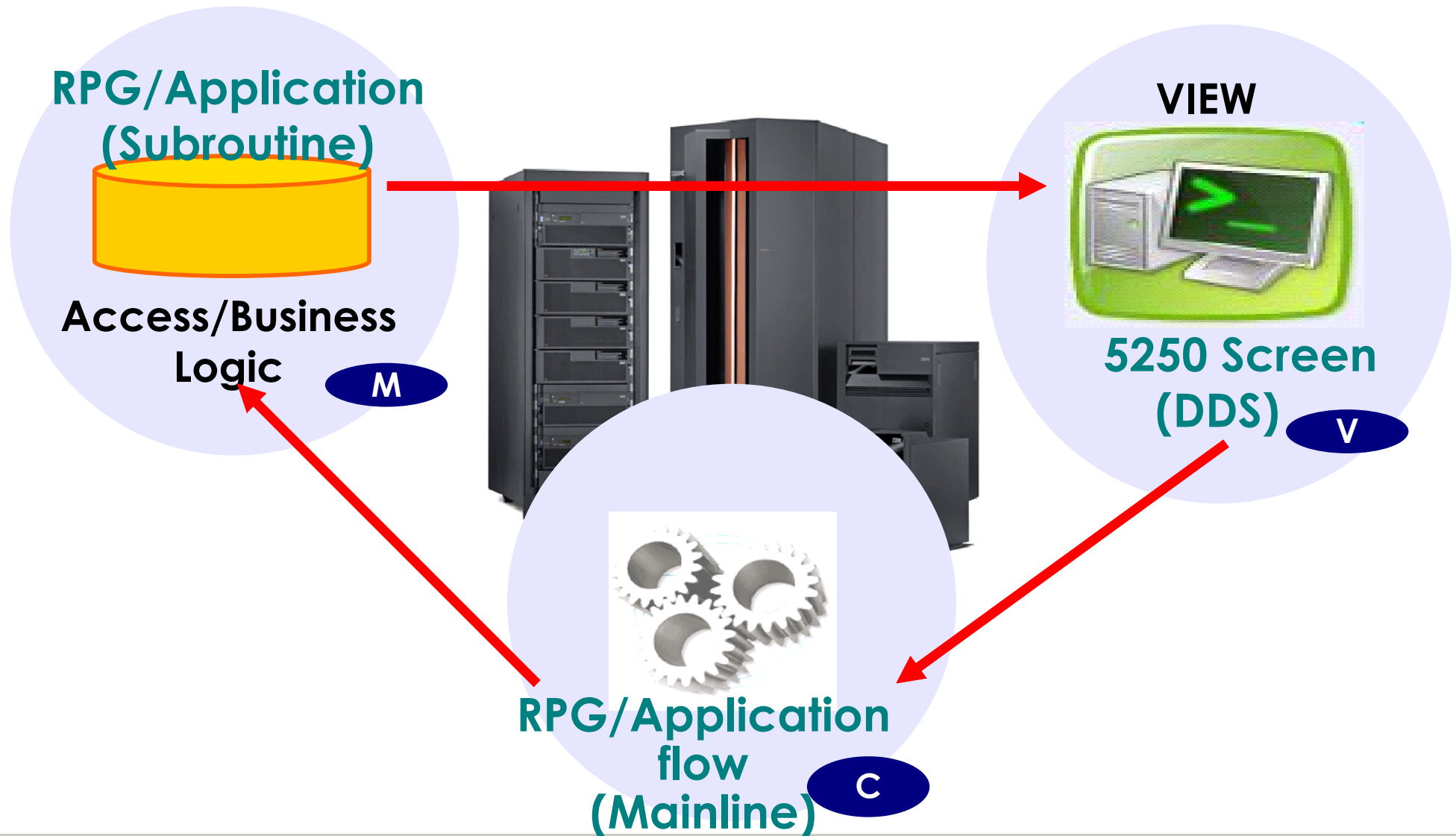
Confession

- **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

Initializize MVC

Front controller to action controller

Frontend Generators	
ZTemplate	ZForm
ZAjax	ZGetText
ZPdf	

Data	
ZDbTable	ZActiveRecord
ZSearch	ZInputFilter
ZXQuery	ZDbSelect
ZRequest	ZXmlRepository

HTTP Request →

Front Controller (URI Mapping)

Plug-ins

Logging	Security
Statistics	

→ **Request Routing** →

Action Controller (Action Logic)

Actions

Display	Query
Add	Remove

→ **Business Logic** →

Component Library

Web Services	
ZRest (Google, Yahoo!, eBay, Amazon)	ZSoap, ZWsdl
ZXmlRpc	ZFeed

User	
ZAuth	ZAcl
ZSession	

Misc	
ZNavigation	ZConfiguration
ZMail	ZLog
ZUri	ZComponent

Documentation	
phpDoc	Manual (DocBook)

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

http request → Bootstrap: index.php → Front Controller → Controller1 action1() action2() / Controller2 action1() action2()
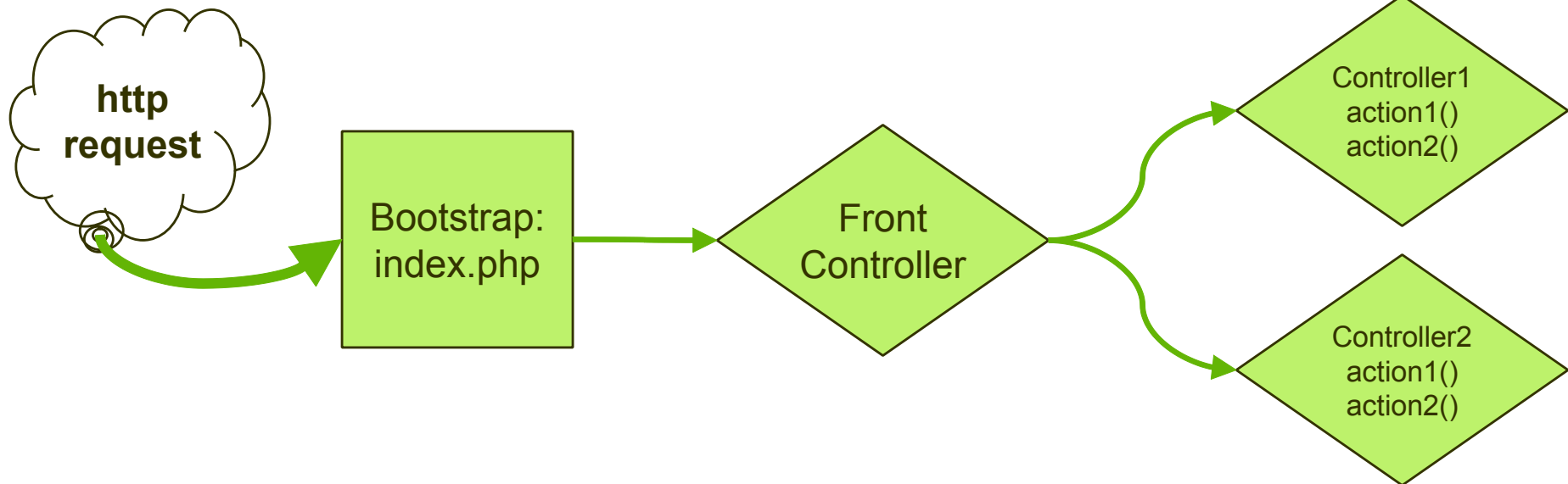
All requests routed through index.php in doc root

Document root is the only public folder.

index.php:

- initializes application

- instantiates Front Controller

C:\wamp\apps\zfdemo\public

On i5, I prefer to edit the Apache .conf file directly rather than add .htaccess files.
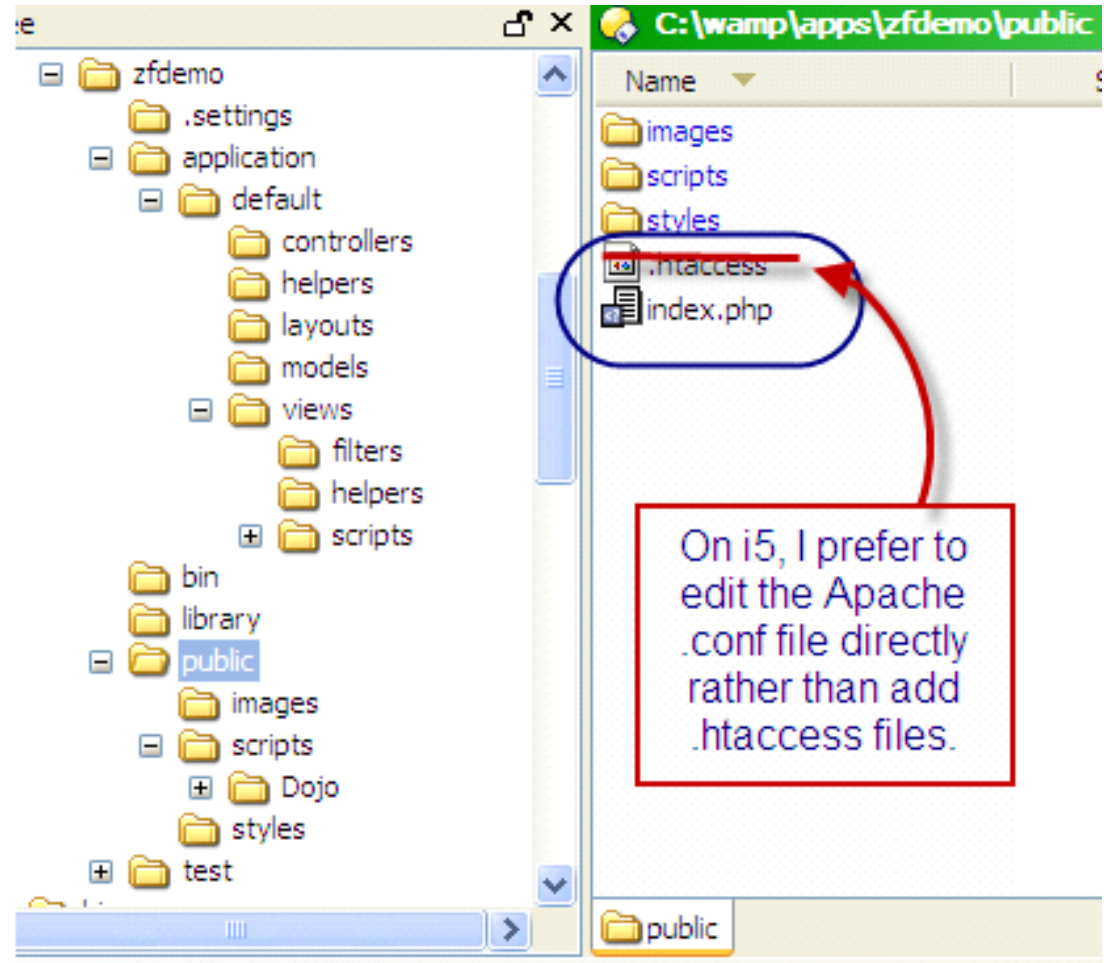
Apache configuration

Most tutorials suggest .htaccess, but I prefer to use the main Apache config file:
/www/zendsvr/conf/httpd.conf

```
Listen 80
Listen 10088
RewriteEngine on

NameVirtualHost 10.11.12.13:80
<VirtualHost 10.11.12.13:80>
    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
<?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();
```

Model-View-Controller
(Good structure for your application)

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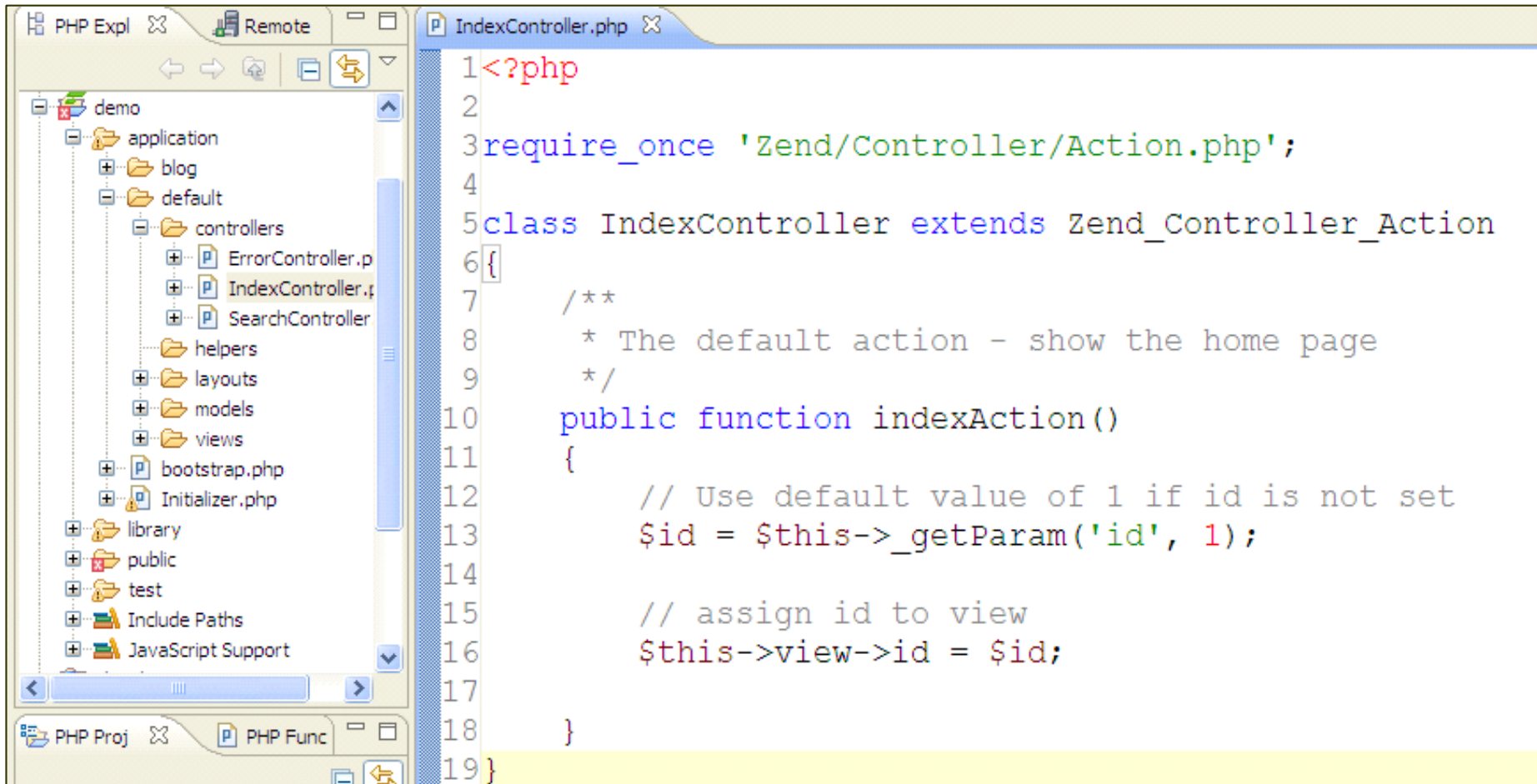
- **Action methods in each controller class handle requests**

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

  Default: indexAction()

  - **Named like *action*Action()**
    - **Example: If *action* is "edit" then method is editAction()**

Controller example

```php
<?php

require_once 'Zend/Controller/Action.php';

class IndexController extends Zend_Controller_Action
{
    /**
     * The default action - show the home page
     */
    public function indexAction()
    {
        // Use default value of 1 if id is not set
        $id = $this->_getParam('id', 1);

        // assign id to view
        $this->view->id = $id;

    }
}
```

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

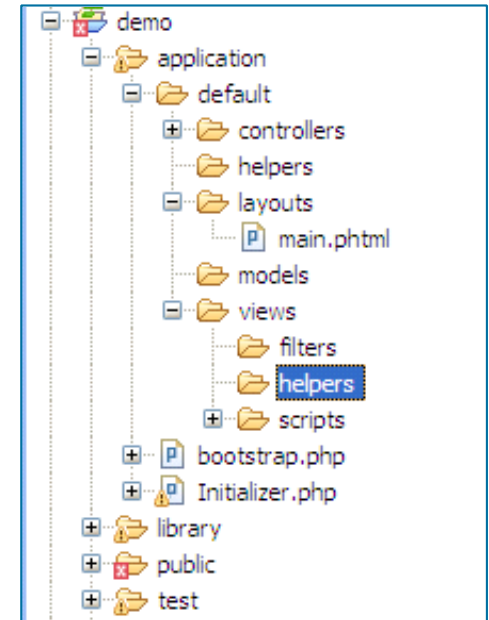**You can plunk HTML right into the view script and replace literals with PHP echo statements:**

- `<?php echo $this->productNum ?>`

ZF provides smart defaults

- The `$this->escape()` view helper uses PHP's htmlentities() function, recommended by most security experts.

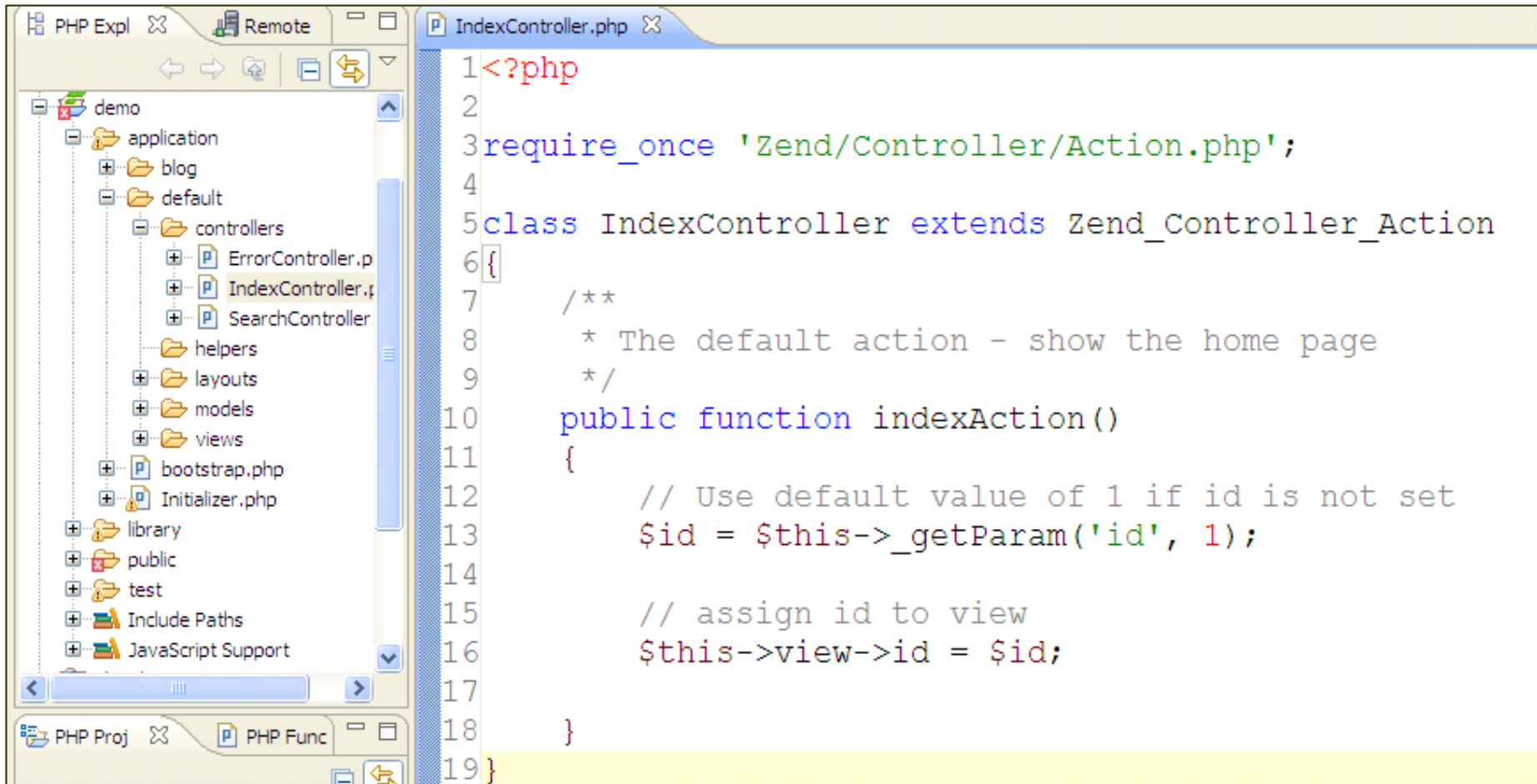# My own view helper: TitleCase.php

```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;
    }
}
```

```
demo
  application
    default
      controllers
      helpers
      layouts
        main.phtml
      models
      views
        filters
        helpers
        scripts
    bootstrap.php
    Initializer.php
  library
  public
  test
```
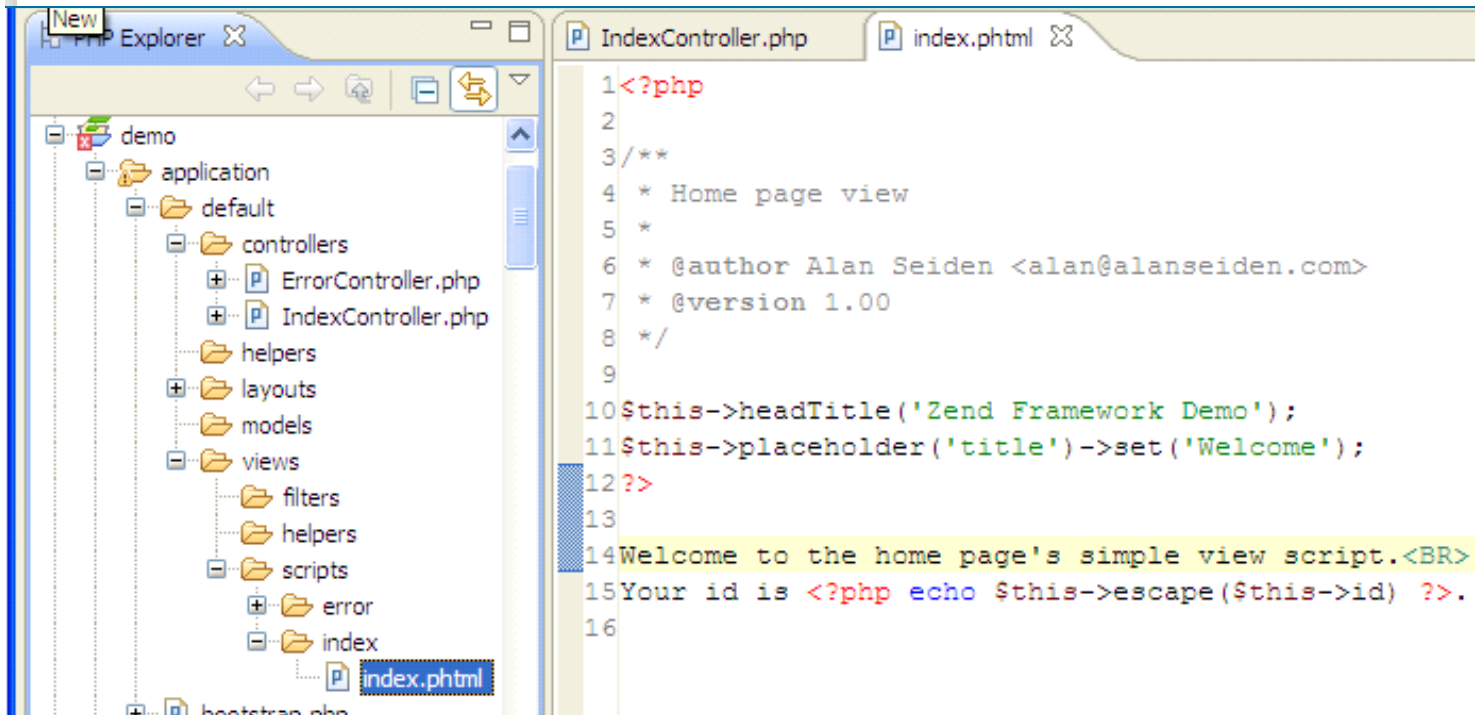
Usage:

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

// Mozilla Firefox
```

Controller (again)…leads to view


```php
<?php

require_once 'Zend/Controller/Action.php';

class IndexController extends Zend_Controller_Action
{
    /**
     * The default action - show the home page
     */
    public function indexAction()
    {
        // Use default value of 1 if id is not set
        $id = $this->_getParam('id', 1);

        // assign id to view
        $this->view->id = $id;

    }
}
```

View script automatically rendered
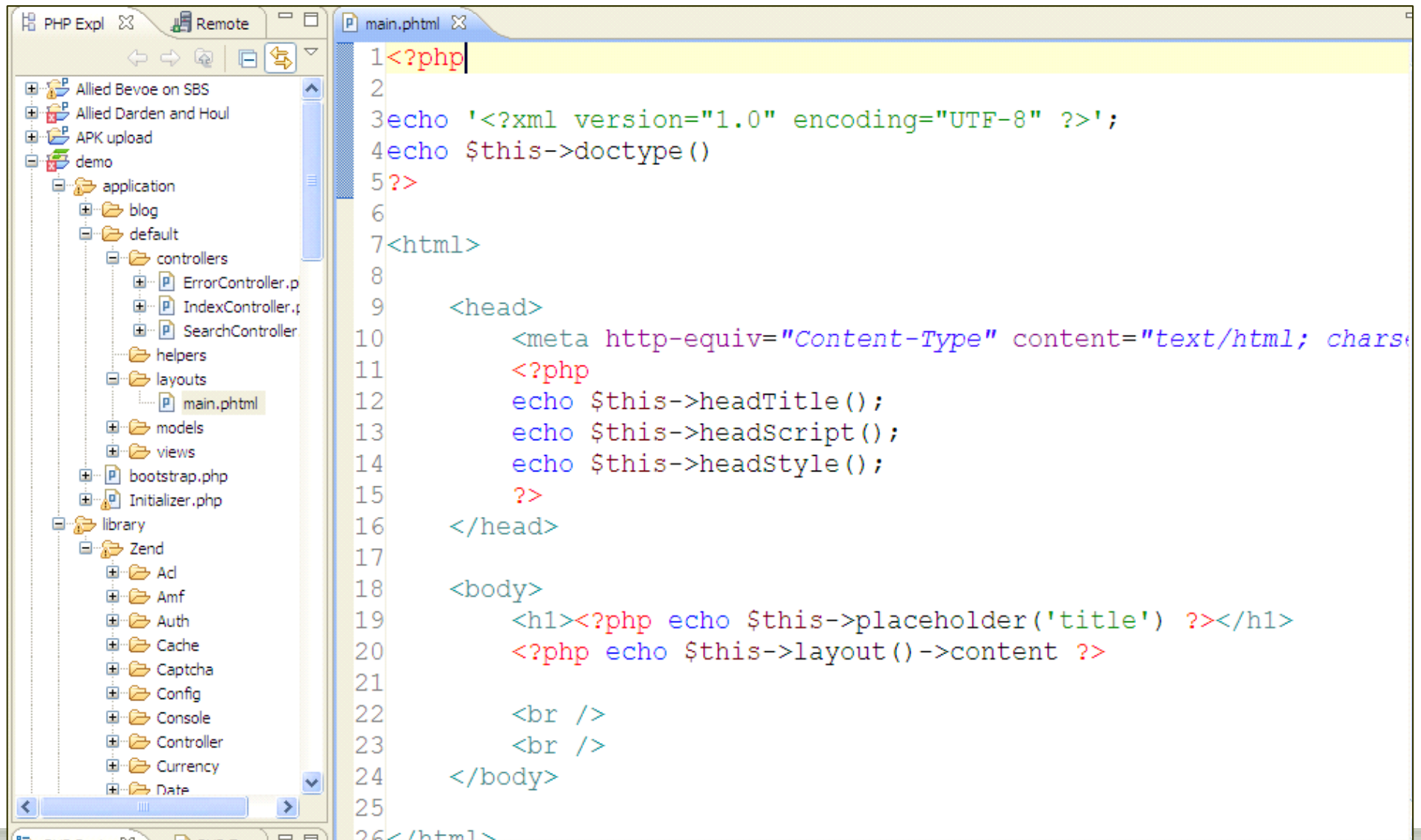
Zend_Layout

- **Gives your site a consistent look while retaining flexibility**

- **Two-step view pattern**

- **Placeholders useful for setting javascript, titles, other variable data**

Zend_Layout code


```php
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; chars
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

```php
// 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)
```
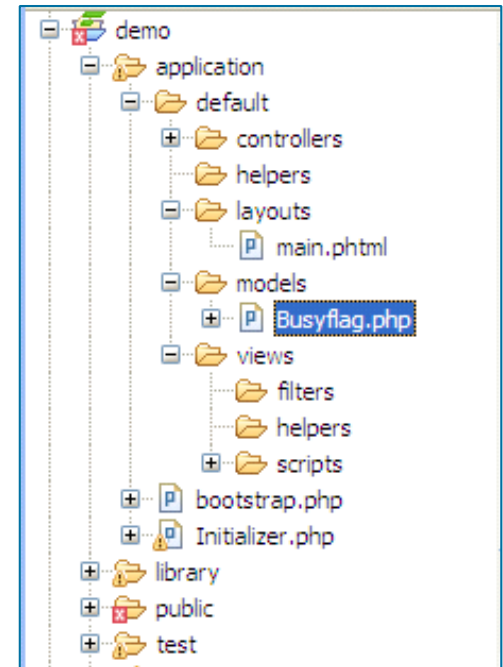
```
demo
  application
    default
      controllers
      helpers
      layouts
        main.phtml
      models
        Busyflag.php
      views
        filters
        helpers
        scripts
    bootstrap.php
    Initializer.php
  library
  public
  test
```
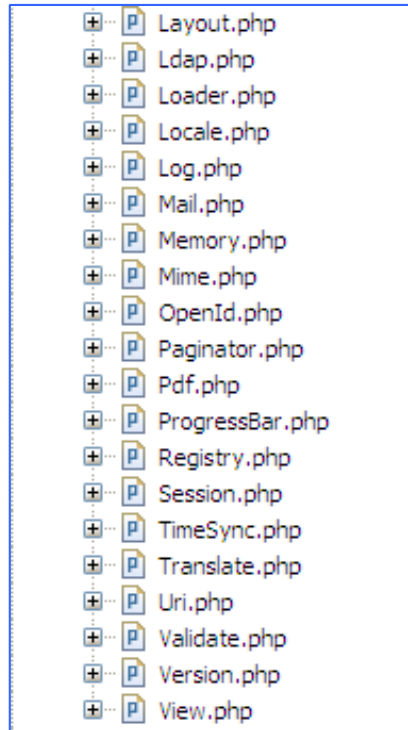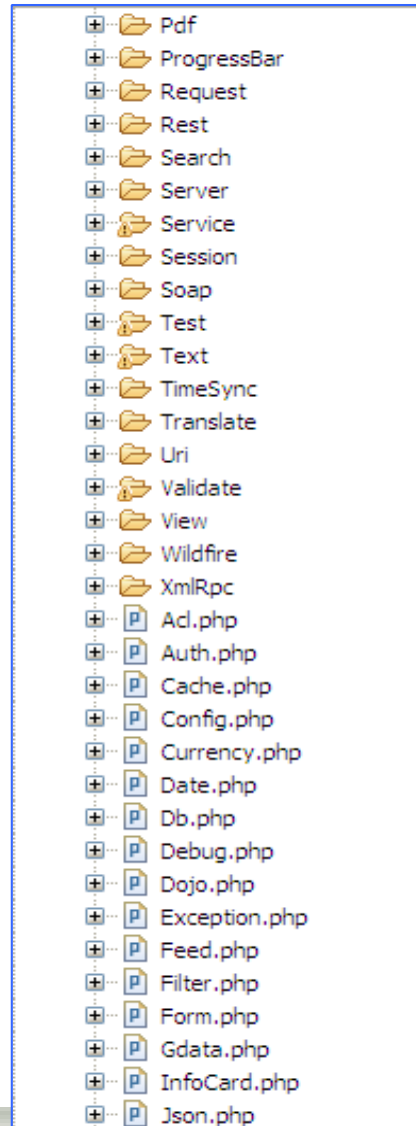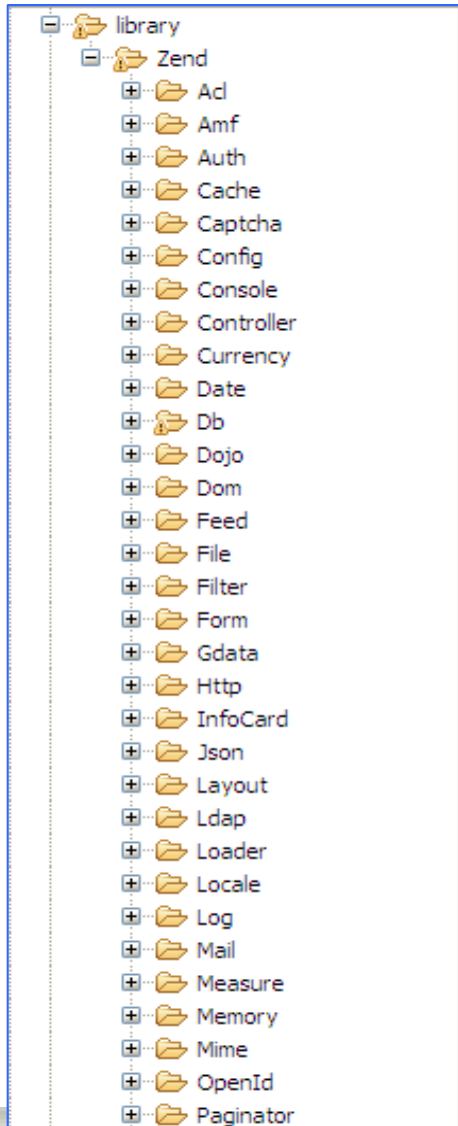
```php
// usage (from a preDispatch front controller plugin)

$busyFlag = new Busyflag();

if (!$busyFlag->isSiteUp()) {

    // Take user to "site down" page.

} //(if (!$busyFlag->isSiteUp()))
```

Components included

Library of Zend components

```
library
  Zend
    Acl
    Amf
    Auth
    Cache
    Captcha
    Config
    Console
    Controller
    Currency
    Date
    Db
    Dojo
    Dom
    Feed
    File
    Filter
    Form
    Gdata
    Http
    InfoCard
    Json
    Layout
    Ldap
    Loader
    Locale
    Log
    Mail
    Measure
    Memory
    Mime
    OpenId
    Paginator
```

```
    Pdf
    ProgressBar
    Request
    Rest
    Search
    Server
    Service
    Session
    Soap
    Test
    Text
    TimeSync
    Translate
    Uri
    Validate
    View
    Wildfire
    XmlRpc
    Acl.php
    Auth.php
    Cache.php
    Config.php
    Currency.php
    Date.php
    Db.php
    Debug.php
    Dojo.php
    Exception.php
    Feed.php
    Filter.php
    Form.php
    Gdata.php
    InfoCard.php
    Json.php
```

```
    Layout.php
    Ldap.php
    Loader.php
    Locale.php
    Log.php
    Mail.php
    Memory.php
    Mime.php
    OpenId.php
    Paginator.php
    Pdf.php
    ProgressBar.php
    Registry.php
    Session.php
    TimeSync.php
    Translate.php
    Uri.php
    Validate.php
    Version.php
    View.php
```

Reminder:

Zend/Db.php = Zend_Db

Zend/Db/Table.php = Zend_Db_Table

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

```php
// 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 results

Implementation of Product Id field

```php
// 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)
```

Zend_Db
database access

Database access with Zend_Db

Zend_Db can create SQL for you

- Fewer long, error-prone concatenated 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

```php
$driverOptions = array('i5_libl' => 'MYLIB1, MYLIB2',
                       'i5_naming => DB2_I5_NAMING_ON);

$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'];
}
```

Zend_PDF

```php
<?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**
- **StrikeIron**
- **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

```php
$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/ib:
platform/
:: 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
Digg - 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

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
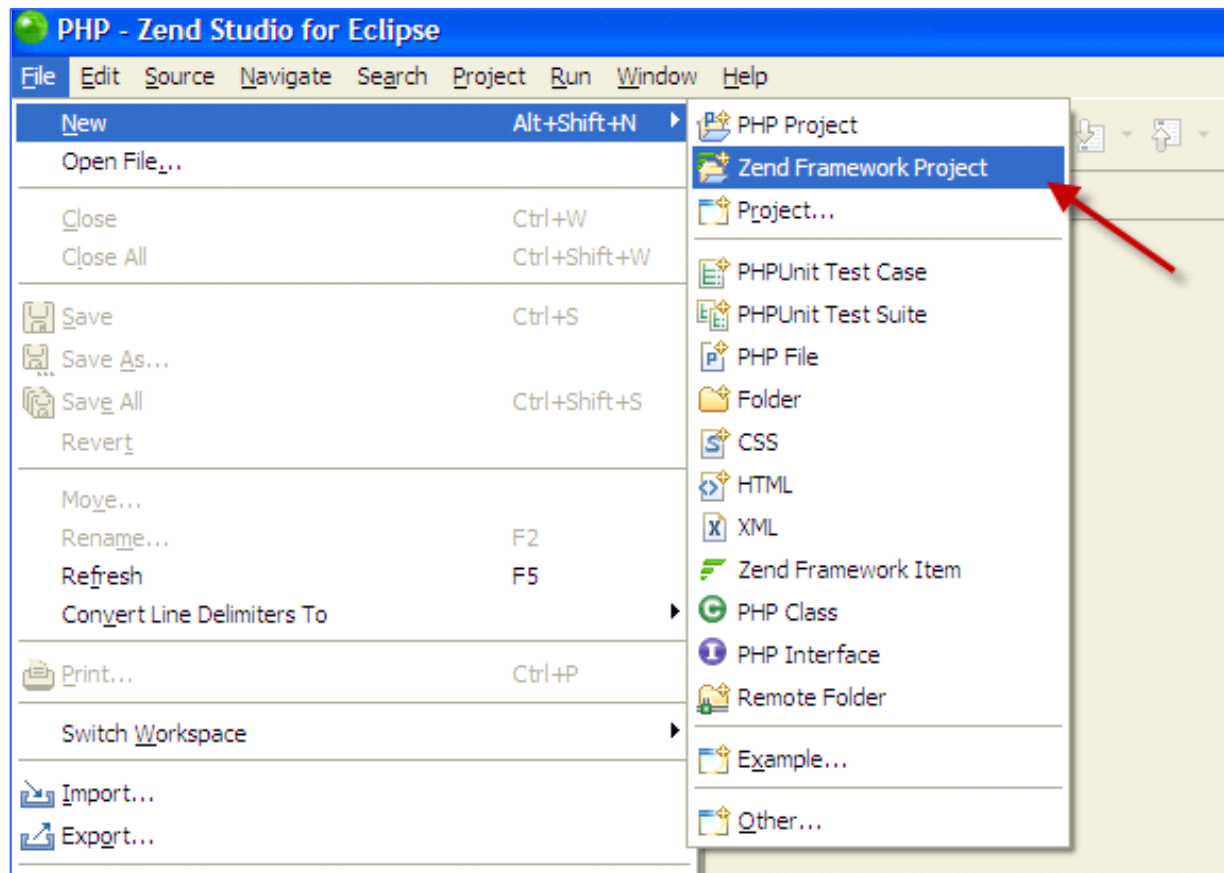
Resources: online

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
    - Search on Meetup.com
  - Subscribe to Zend's ZF support if it's a mission-critical app

Questions and thanks

Blog/articles: http://alanseiden.com
Contact: aseiden@sbsusa.com, 201-327-9400
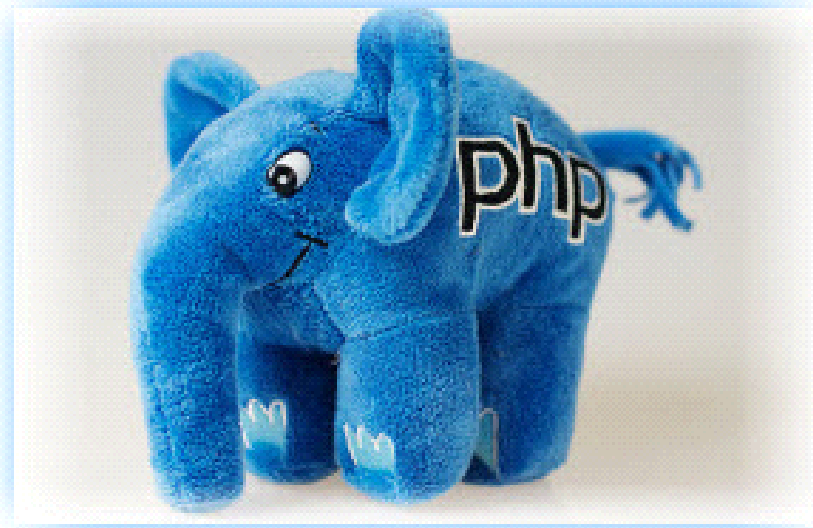
Free PHP/i tips: write to **aseiden@sbsusa.com**
with subject: "PHP tips."
Or write "PHP tips" on a business card today