Zend Framework 1.0 – An Overview

Bill Karwin
Product Engineering Manager,
Zend Technologies

Now, the world's most popular web programming language gets even better… A loosely-coupled framework with a flexible architecture that lets you easily build modern web applications and web services.

Zend Framework 1.0 – An Overview

- Introduction to Zend Framework
- Example Zend Framework MVC application
- Benefits of Zend Framework
- Roadmap for Zend Framework future

What is the Zend Framework?

- PHP 5 library for web development productivity
- Open source
  - New BSD license is business-friendly
  - Free for development and distribution
  - CLA process assures that the code is free of legal issues
- Class library – over 150,000 lines of code
- Documentation – over 500 pages
- Quality & testing – over 4,200 unit tests
  - 84%+ code coverage
  - Deployed on many websites already

Zend Framework philosophy

- "Extreme simplicity"
  - Easy solutions for the 80% most commonly-used functionality for web applications
  - Extensibility enables easy customization, to solve the remaining 20%
  - No complex XML configuration files
- Good object-oriented and agile practices
  - Use-at-will architecture
  - Design for extensibility
  - Frequent testing
  - Frequent interaction with user community

Zend Framework quality process

1. Say what you're going to do
   - Proposal process
2. Do it
   - Object-oriented software development
   - Emphasis on unit tests
   - We encourage test-driven development (TDD) as well
3. Verify it matches what you said
   - Frequent and thorough testing with PHPUnit
   - Open-source development and community review

What's in the Zend Framework?

- MVC
- Database
- I18N
- Authentication
- Web Services
- Mail, Formats, Search
- Utility

- Zend_Controller
  - Front controller
  - Routers
  - Action handling
  - Plugins and Helpers
  - Request
  - Response
- Zend_View
  - PHP template class
  - Helpers
  - Filters
- Zend_Json (helps Ajax)

What's in the Zend Framework?

- MVC
- Database
- I18N
- Authentication
- Web Services
- Mail, Formats, Search
- Utility

- Zend_Db_Adapter
  - IBM DB2
  - Oracle
  - MS SQL Server
  - PostgreSQL
  - MySQL
  - SQLite
- Zend_Db_Profiler
- Zend_Db_Select
- Zend_Db_Table
  - Zend_Db_Table_Rowset
  - Zend_Db_Table_Row
  - like ActiveRecord

What's in the Zend Framework?

- MVC
- Database
- I18N
- Authentication
- Web Services
- Mail, Formats, Search
- Utility

- Zend_Locale
- Zend_Translate
  - Array
  - Csv
  - Gettext
  - Qt
  - Tmx
  - Xliff
- Zend_Date
- Zend_Measure

What's in the Zend Framework?

- MVC
- Database
- I18N
- Authentication
- Web Services
- Mail, Formats, Search
- Utility

- Zend_Auth
  - DbTable
  - Digest
  - Http
- Zend_Session
  - Persist identity, other data
- Zend_Acl
  - Manage roles, privileges

What's in the Zend Framework?

- MVC
- Database
- I18N
- Authentication
- Web Services
- Mail, Formats, Search
- Utility

- Zend_Http_Client
- Zend_Rest_Client
- Zend_Service
  - Akismet
  - Amazon
  - Audioscrobbler
  - Delicious
  - Flickr
  - Simpy
  - StrikeIron
  - Yahoo
- Zend_Feed (RSS and Atom)
- Zend_Gdata (Google Data API)
- Zend_XmlRpc_Client

What's in the Zend Framework?

- MVC
- Database
- I18N
- Authentication
- Web Services
- Mail, Formats, Search
- Utility

- Zend_Mail
- Zend_Mime
  - Read or send email

- Zend_Pdf
  - Read, edit & create PDF documents

- Zend_Search_Lucene
  - Search engine
  - Apache Lucene compatible

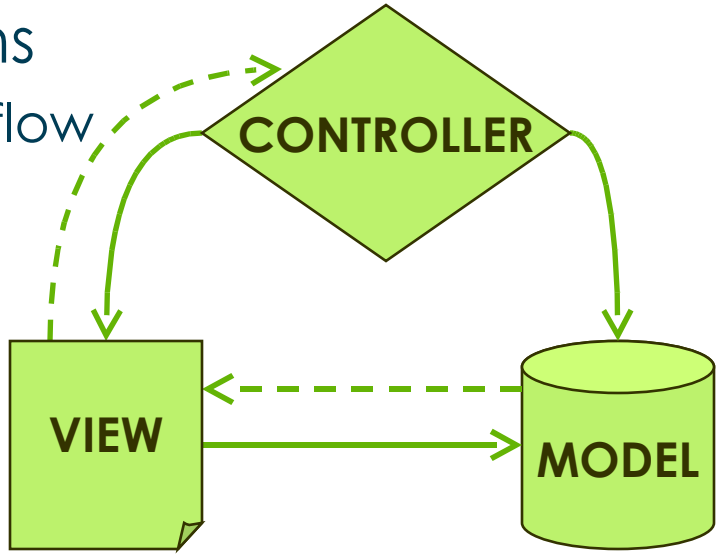
What's in the Zend Framework?

- MVC
- Database
- I18N
- Authentication
- Web Services
- Mail, Formats, Search
- Utility

- Zend_Cache
- Zend_Config
- Zend_Console_Getopt
- Zend_Filter
- Zend_Filter_Input
- Zend_Loader
- Zend_Log
- Zend_Memory
- Zend_Registry
- Zend_Validate

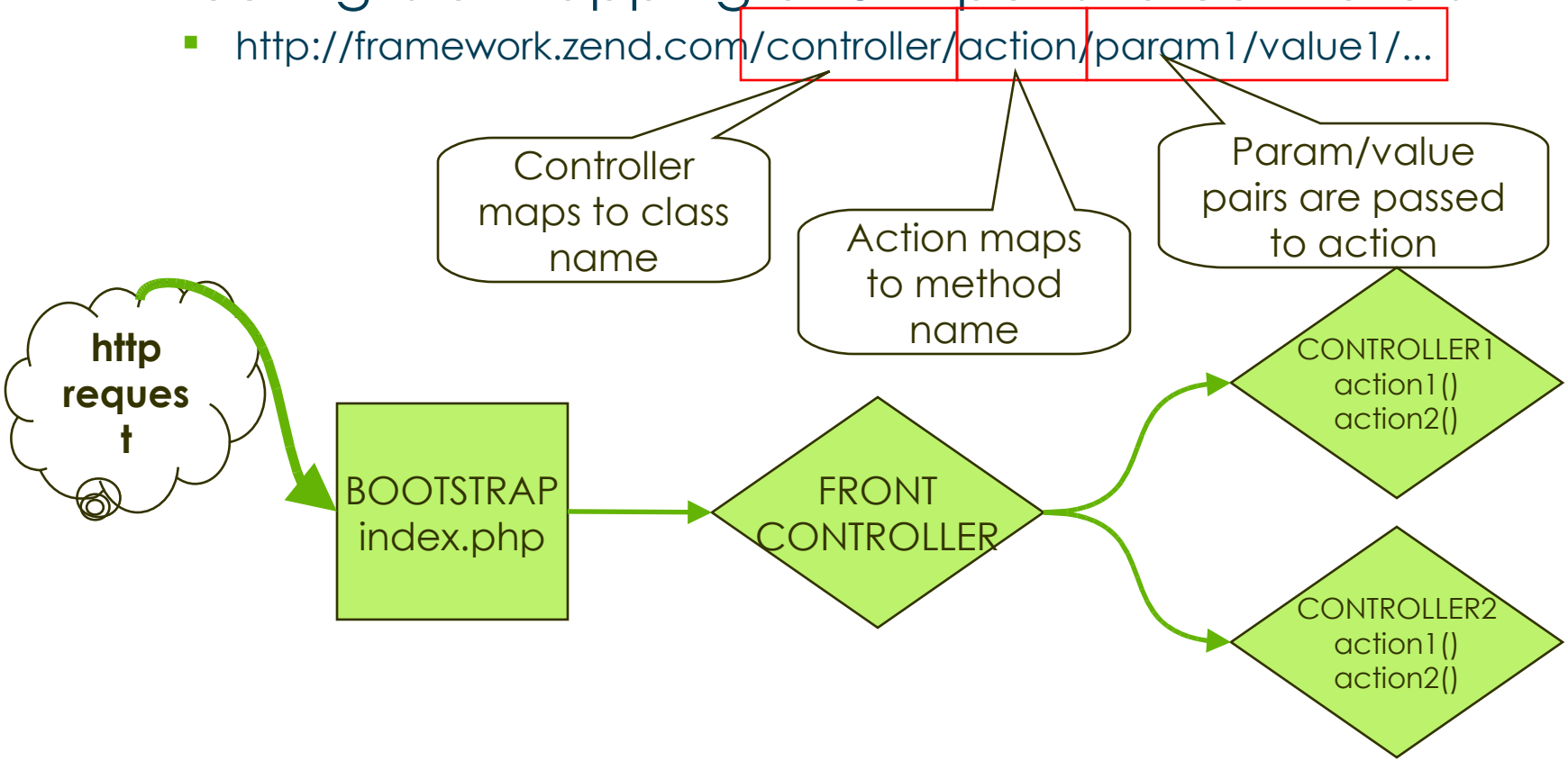
What is the MVC component?

- The heart of ZF web applications
  - Best practice for application workflow
  - Design pattern that dates back to Smalltalk, circa 1979
- Model: data provider
  View: user interface
  Controller: request processor
- Simple solution in most applications
  - Sensible defaults are built in
  - Flexible and extensible
  - Supports advanced applications



Features of MVC

- The *Front Controller* routes requests to controllers
- Routing is a mapping of URL parts to controllers
  - http://framework.zend.com/controller/action/param1/value1/...



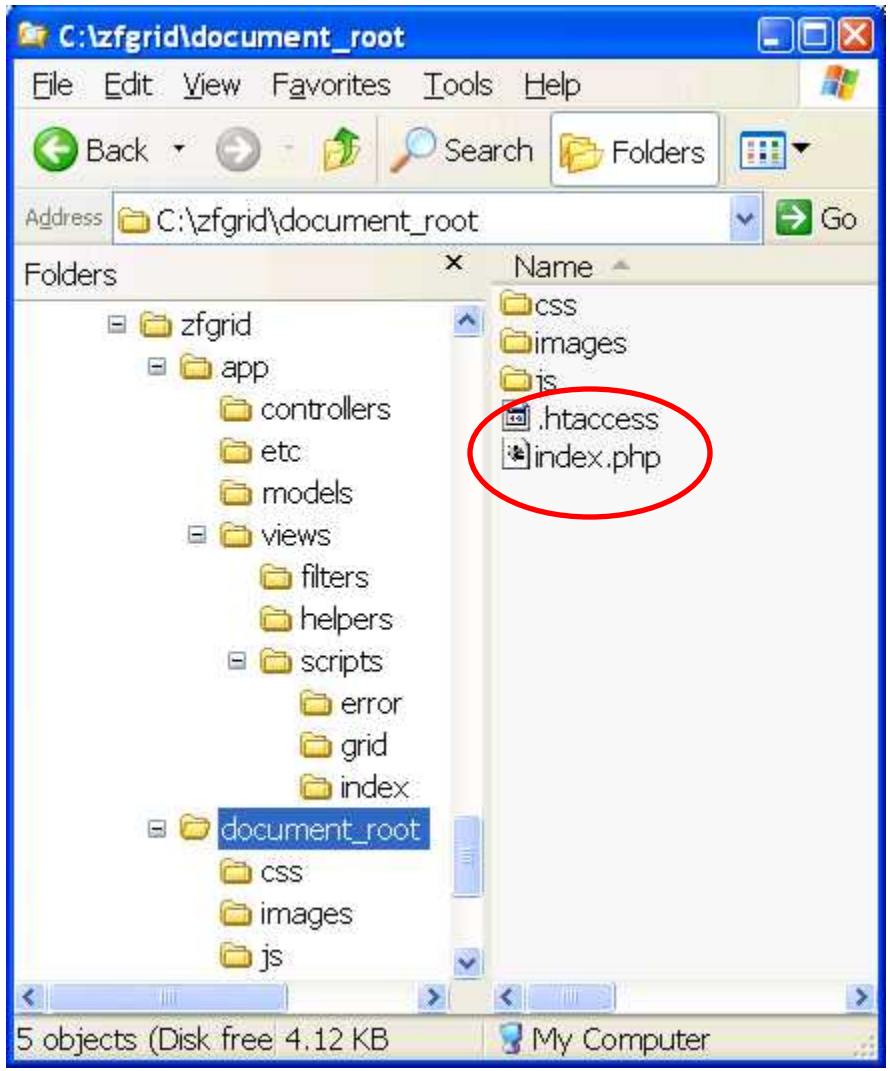
Features of MVC

- Declare custom routing rules
  - Not limited to "controller/action/param" format
- Optional Controller Plugins, Action Helpers, and View Helpers
  - ErrorHandler plugin handles exceptions, 404 errors, etc.
  - FlashMessenger, Redirector, ViewRenderer helpers
  - Output common HTML elements in views
- Extensible interfaces
  - Write your own plugins and helpers

Example MVC application

How to use MVC: bootstrap

- Document root
  - **.htaccess**
    enables Apache *RewriteEngine*, which redirects requests to bootstrap script
  - **index.php**
    instantiates the Front Controller, which routes requests to the correct controller



How to use MVC: .htaccess

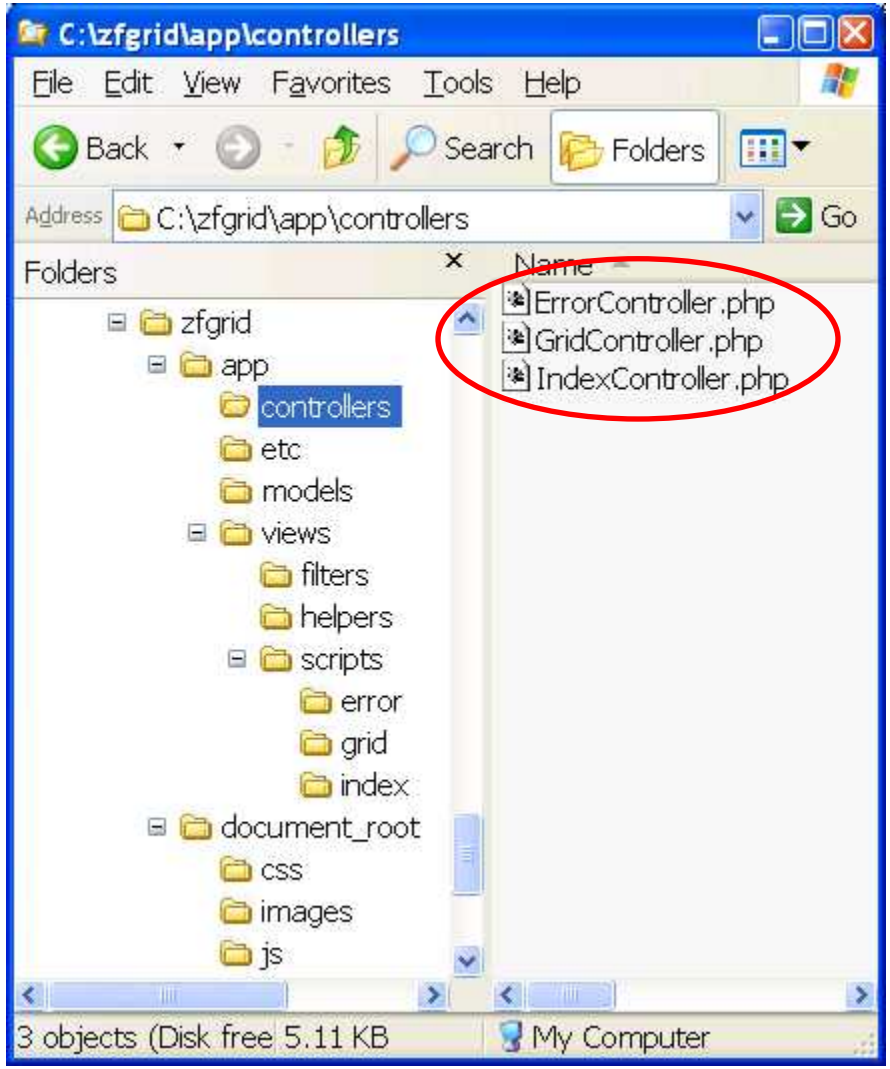
```
RewriteEngine on
# funnel all requests to index.php
# except requests for static resources
RewriteRule !\.(js | ico | gif | jpg | png | css)$ index.php
```

How to use MVC: index.php

```
<?php
error_reporting( E_ALL | E_STRICT );

// optional but handy
require_once 'Zend/Loader.php';
Zend_Loader::registerAutoload();

// where is the rest of the application?
$appDir = dirname(dirname(__FILE__)) . '/app';

// convenience method to dispatch
Zend_Controller_Front::run("$appDir/controllers");
```

How to use MVC: controllers

- Controller classes handle groups of request URLs
  - http://zend.com/**controller**/action
  - The default controller class is "IndexController"
- Action methods in each controller class handle individual requests
  - http://zend.com/controller/**action**
  - The default action method is "indexAction()"



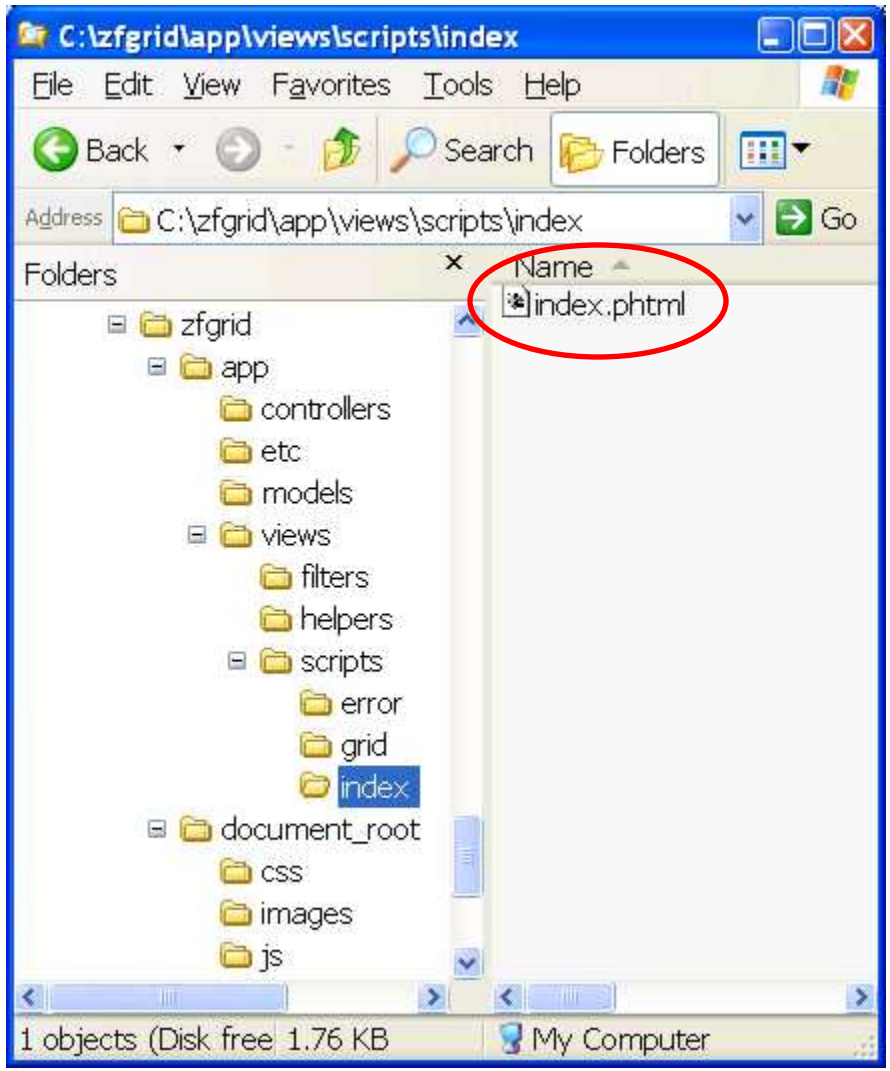
How to use MVC: IndexController.php

```
class IndexController extends Zend_Controller_Action
{
    protected $_db;

    public function init()
    {
        $params = array('dbname' => 'xxxx',
                        'username' => 'xxxx', 'password' => 'xxxx');
        $this->_db = Zend_Db::factory('Mysqli', $params);
    }

    public function indexAction()
    {
        $this->view->tableList = $this->_db->listTables();
    }
}
```

How to use MVC: views

- Views are PHP-based script templates to present data
- Views should contain only display logic, not business logic

Tip: if two View scripts display the same data, you should not need to duplicate code to prepare the data; you should write that code in a Model class



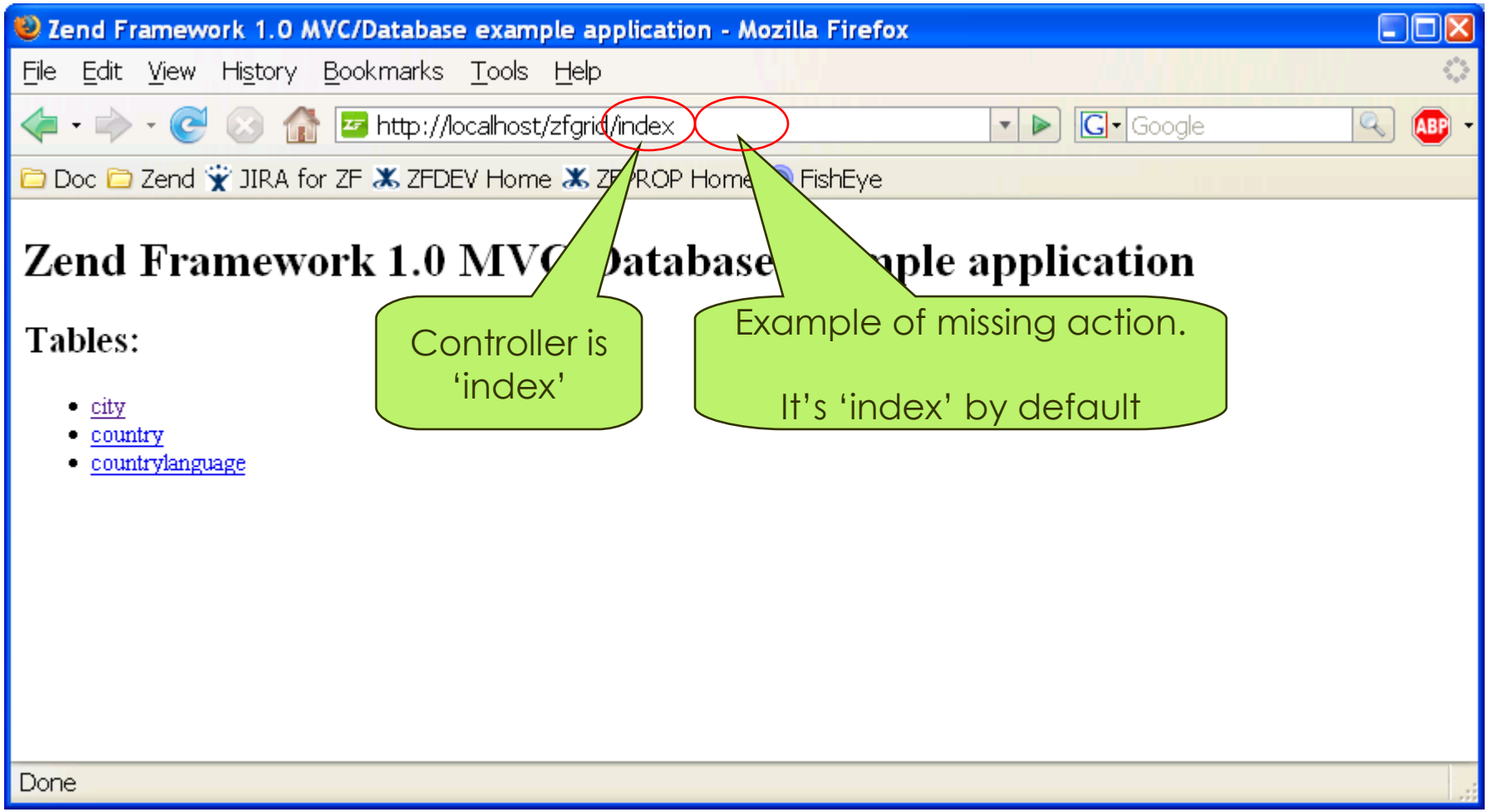
How to use MVC: index.phtml

```
1.  <html>
2.      <head>...</head>
3.      <body>
4.          <h1>Zend Framework 1.0 MVC/Database example a
5.          <? if ($this->tableList): ?>
6.          <h2>Tables:</h2>
7.          <ul>
8.              <? foreach ($this->tableList as $tableName): ?>
9.              <li>
10.                 <a href="/zfgrid/grid/show/table/<?= $tableName ?>">
11.                     <?= $tableName ?>
12.                 </a>
13.             </li>
14.             <? endforeach; ?>
15.         <ul>
16.         <? endif; ?>
17.     </body>
18. </html>
```

Loop over list of tables

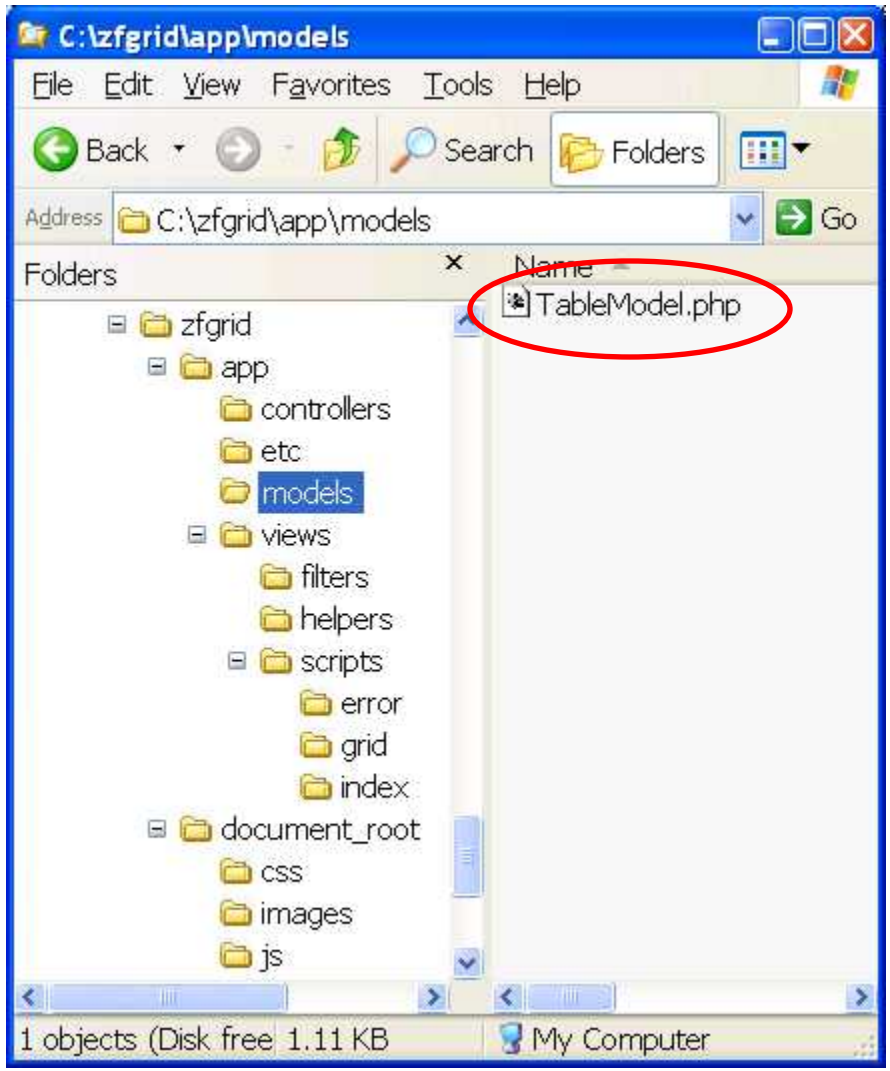
Link to grid controller contains 'table' param

How to use MVC: output of index.phtml



How to use MVC: models

- Model classes contain business logic to prepare complex data for presentation
- Often employs database queries, but it could fetch an RSS feed, etc.
- A controller action may use multiple models



How to use MVC: TableModel.php

```
class TableModel extends Zend_Db_Table_Abstract
{
    /*
     * No code is necessary in this case.
     * The Controller will tell this object which table to read.
     * All methods are inherited from the parent class.
     */
}
```

How to use MVC: GridController.php

```
1.  class GridController extends Zend_Controller_Action
2.  {
3.      public function init()
4.      {
5.          $tableName = $this->_getParam('table', false);
6.          if ($tableName === false) {
7.              $this->_redirect('/index');
8.          }
9.          $this->_model          = new TableModel(array('name' => $tableName));
10.     }

12.     public function showAction()
13.     {
14.         $order                 = $this->_getParam('order', null);
15.         $count                 = (int) $this->_getParam('count', 20);
16.         $offset                = (int) $this->_getParam('offset', 0);
17.         $this->view->tableInfo = $this->_model->info();
18.         $this->view->rowset    = $this->_model->fetchAll(
                                       null, $order, $count, $offset);
19.     }
20. }
```

Default value if 'table' param is not present

Create a Model object

Call methods to get data from the Model object

Pass data to the View object

How to use MVC: index.phtml

```
1.  <table cellspacing="0" cellpadding="2">
2.      <tr class="rowHead">
3.          <? foreach ($this->tableInfo['cols'] as $columnName): ?>
4.          <th>
5.              <a href="/zfgrid/grid/show/table/<?= $this->tableInfo['name'] ?
    >/order/<?= $columnName ?>">
6.              <?= $columnName ?></a>
7.          </th>
8.          <? endforeach; ?>
9.      </tr>
10.     <?php    $i = 0;    $stripeClass = array('rowEven', 'rowOd
11.     <? foreach ($this->rowset as $row): ?>
12.     <tr class="<?= $stripeClass[$i++%2] ?>">
13.         <? foreach ($this->tableInfo['cols'] as $columnName): ?>
14.         <td>
15.             <?= $this->escape($row->$columnName) ?>
16.         </td>
17.         <? endforeach; ?>
18.     </tr>
19.     <? endforeach; ?>
20. </table
```

Loop over list of column names

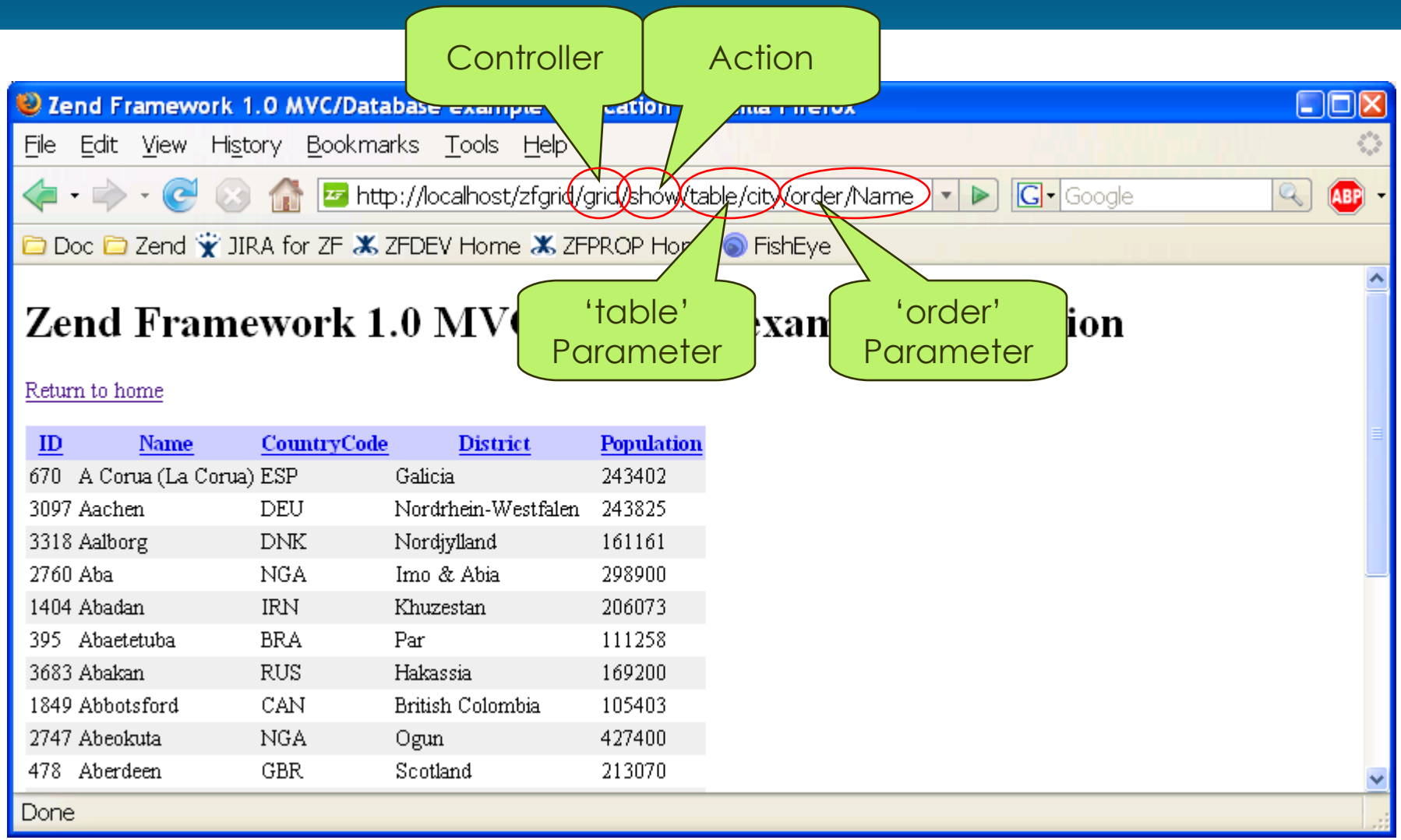
Link contains 'order' parameter to sort by column

Loop over rowset

Loop over columns in each row

Output data content

How to use MVC: output of grid/show.phtml



Examples of some Zend Framework benefits

- Improving application security
- Encouraging best practices
- Avoiding reinventing the wheel

3 tools ZF offers to improve application security

1. Zend_Filter_Input
2. Zend_Auth
3. Zend_View

- Processes values of form fields
- Declares rules for filtering and validating input data
- Serves as a "cage" for data; only valid data come out
- Provides method for escaping values, to make them safer for HTML output

3 tools ZF offers to improve application security

1. Zend_Filter_Input
2. Zend_Auth
3. Zend_View

- Abstract interface to login credential storage
- Provides adapters for several directory solutions
  - Digest
  - Http
  - Database table
  - others
- Write your own custom directory adapters

3 tools ZF offers to improve application security

1. Zend_Filter_Input
2. Zend_Auth
3. Zend_View

- Provides a method to format PHP objects more safely for HTML:

```
<?= $this->escape($this->var); ?>
```

4 ways ZF encourages best practices

1. Object-oriented programming
2. Testing
3. Documentation
4. Extremely simple

- Most classes are reusable and extensible, allowing you to write less code
- User-defined extensions, plugins, and adapters implement interfaces and should be well-designed classes
- Uses many other design patterns

4 ways ZF encourages best practices

1. Object-oriented programming
2. Testing
3. Documentation
4. Extremely simple

- MVC Request and Response can be simulated
- Built-in testability for Controllers
- Extensive collection of PHPUnit test suites serves as useful example

4 ways ZF encourages best practices

1. Object-oriented programming
2. Testing
3. Documentation
4. Extremely simple

- Framework development process requires complete manual coverage for every component
- Framework coding standards include complete API docblocks to support reference documentation and development tools

4 ways ZF encourages best practices

1. Object-oriented programming
2. Testing
3. Documentation
4. Extremely simple

- Simple interfaces are better for:
  - Usability
  - Extensibility
  - Testability
  - Maintainability
- Avoid "AntiPatterns"
  - Functional decomposition
  - Spaghetti code
  - Sequential coupling
  - http://www.antipatterns.com/

5 ways ZF helps to avoid reinventing the wheel

1. Zend_Acl
2. Zend_Feed
3. Zend_Log
4. Zend_Cache
5. Zend_Config

- Manages application roles, resources and privileges
- Roles and Resources may be user-defined classes
- Supports rule inheritance
- Supports conditional application of rules

5 ways ZF helps to avoid reinventing the wheel

1. Zend_Acl
2. Zend_Feed
3. Zend_Log
4. Zend_Cache
5. Zend_Config

- Consume RSS feed in a single line of PHP code
- Discover feed links automatically
- Imports feeds from multiple sources
- Provides feed building and posting operations

5 ways ZF helps to avoid reinventing the wheel

1. Zend_Acl
2. Zend_Feed
3. Zend_Log
4. Zend_Cache
5. Zend_Config

- Support advanced yet easy logging operations in your application
- Inspired by log4j
- Supports user-defined log formatting and writing

5 ways ZF helps to avoid reinventing the wheel

1. Zend_Acl
2. Zend_Feed
3. Zend_Log
4. Zend_Cache
5. Zend_Config

- Provides an interface to access persisted data
- Supports tagging, manipulating, iterating, and removing subsets
- Supports multiple cache storage backends (e.g. file, database, memcached, and Zend Platform)

5 ways ZF helps to avoid reinventing the wheel

1. Zend_Acl
2. Zend_Feed
3. Zend_Log
4. Zend_Cache
5. Zend_Config

- Simplifies usage of configuration data for applications
- Provides a property-driven OO interface for accessing data hierarchy
- Supports multiple storage formats (.INI files, XML)
- Supports inheritance of configuration sections
- Optional; Zend Framework is configuration-less

Zend Framework Roadmap

- Zend Framework 1.0.1 – by August 1
- Zend Framework 1.1.0 – by October 8
- Develop online tutorials and examples
- Continue to foster our Web Services ecosystem
- Support digital identity management
- Provide advanced Web Forms solution
- Integrate with development tools, including visual tools like Zend Studio

Questions & Answers

Join the world's largest gathering of the PHP community!

- More than 40 technical sessions led by PHP experts
- Unique networking events with PHP community members
- Exhibit Hall with leading companies and cutting-edge solutions

Register now and save – www.zendcon.com

New! Zend Framework Training – www.zend.com/framework-training

Thanks

http://framework.zend.com/