

Zend Framework Overview

29 February 2008

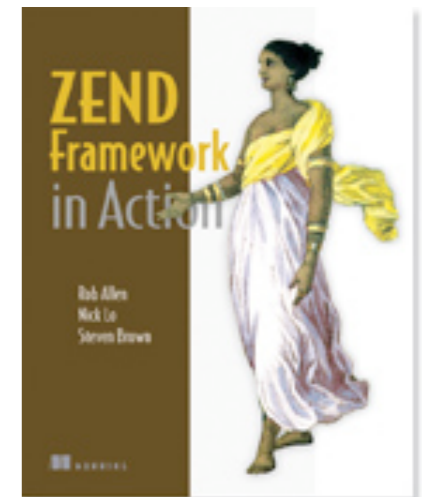
Rob Allen
<http://akrabat.com>

What will I cover?

- Who am I?
- What is Zend Framework?
- Why Zend Framework?
- ZF MVC overview
- Forms overview

Rob Allen?

- PHP developer since 1999
- Contributor to Zend Framework
- Intro ZF tutorial on <http://akrabat.com>
- Book: **Zend Framework in Action**



(coming soon)

What is Zend Framework?

- “Glue” framework: use what you need
- Loosely coupled
- Easy integration with any framework
- PHP 5 only
- Available from <http://framework.zend.com>

Business friendly

- BSD License
- CLA: Free from IP disagreements
- Certification (coming soon)
- Actively maintained by Zend

Why use Zend Framework?

- All the boring bits are done for you!
- Modern design: PHP5, OO
- Solves 80% of the problem space
- Flexible and configurable for the 20%
- Committed to minimizing BC breakage

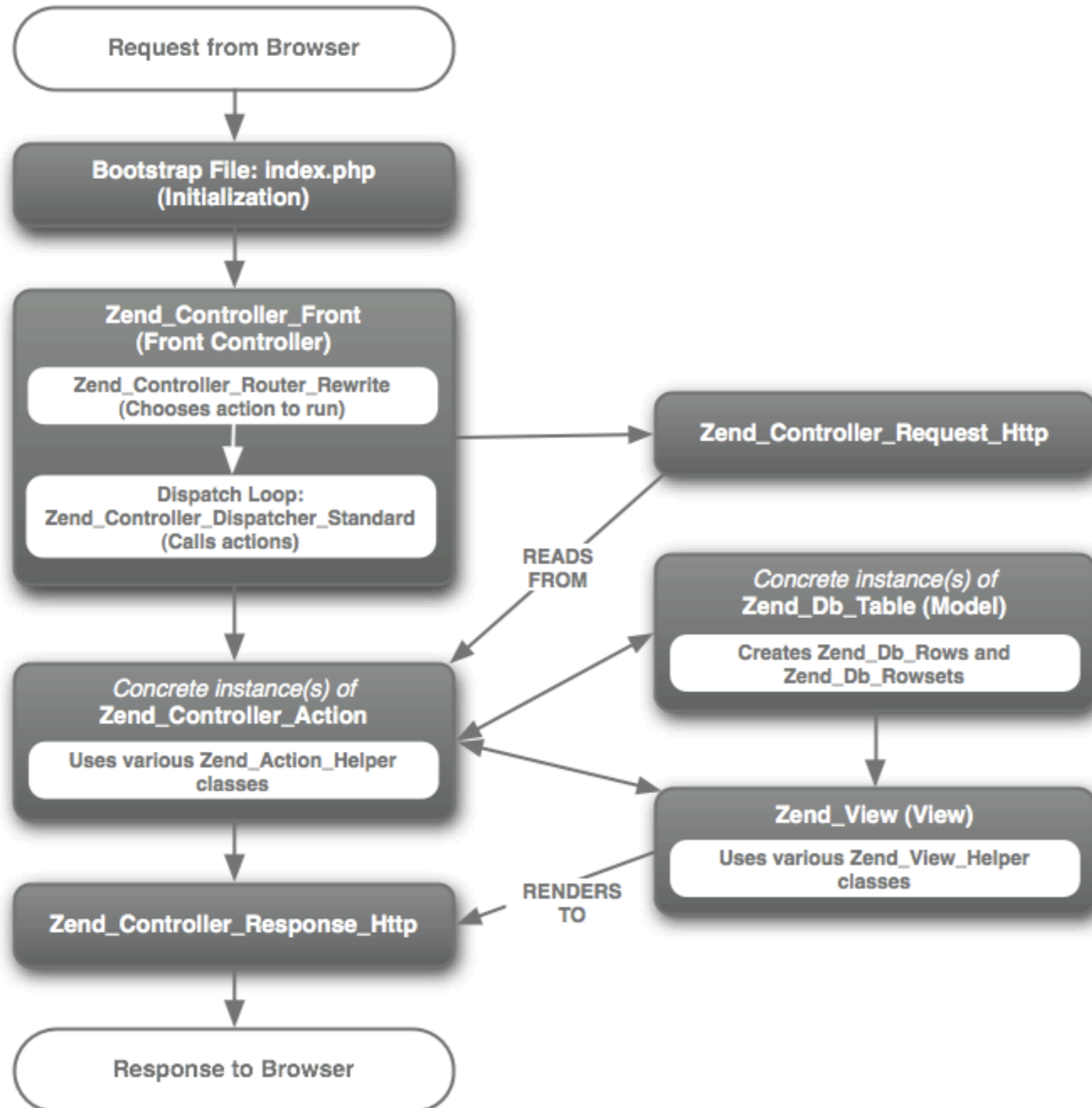
Available resources

- On line documentation
- Community
 - Forums, mailing lists, IRC (#zftalk)
- Books available
 - phplarchitect, Apress, Manning
- Consultancy available from Zend

What's in Zend Framework?

- **Model-View-Controller**
- **Data:** database, json, pdf, search
- **Internationalisation:** locale, translate, date, currency
- **Web services:** feed, http, openid, service(amazon, delicious, google, yahoo, etc)
- **Core services:** auth, acl, cache, config, log, mail, session

ZF MVC flow



Routing

- match() and assemble() URLs
- Easy to create your own route:

e.g. /news/2008/01

```
$route = new Zend_Controller_Router_Route(  
    'news/:year/:month',  
    array(  
        'controller' => 'news',  
        'action'      => 'detail'  
    ),  
);  
$router->addRoute('archive', $route);
```

Action controller

- Does the work!
- Actions grouped into controller class
- Standardised naming:
 - Class name ends in “Controller”
 - Action name ends in “Action”
- Action helpers provide extensibility

```
indexController::indexAction()
```

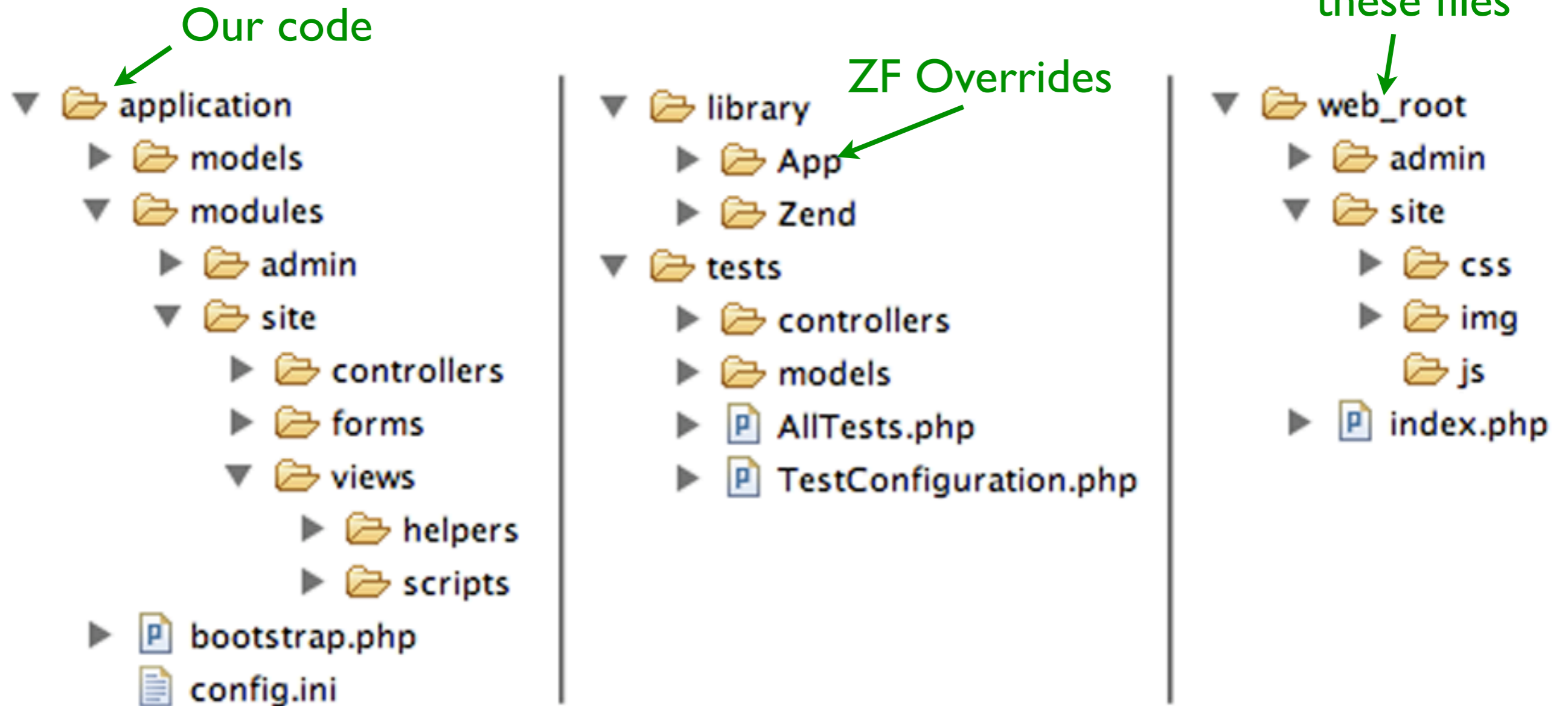
Front controller plugins

- Flex points in controller flow:
 - pre/post Routing
 - pre/post Dispatch loop
 - pre/post Action
- Used by Zend_Layout, ActionStack & ErrorHandler

Action helpers

- Action level plugins
- Reusable functionality
- Useful for automating action controller processes
- Built-in: FlashMessenger, Redirector, ViewRenderer & ContextSwitch

My directory structure



Bootstrap

index.php:

```
include '../application/bootstrap.php';  
$bootstrap = new Bootstrap();  
$bootstrap->runApp();
```

application/bootstrap.php:

```
class Bootstrap  
{  
    public function runApp()  
    {  
        $fc = Zend_Controller_Front::getInstance();  
        $fc->addModuleDirectory('../application/modules/');  
        $fc->registerPlugin(new App_Controller_Plugin_ViewSetup());  
  
        $fc->dispatch();  
    }  
}
```

 **Process request**

Controller code

```
class IndexController extends Zend_Controller_Action
{
    public function indexAction()
    {
        $this->view->title = 'Reasons why Zend Framework rocks:-';

        $items = new Items();
        $this->view->items = $items->fetchAll();
    }
}
```



Retrieve model data
and assign to view

Zend_View

- Controller integration using ViewRenderer
 - Rendered in `postDispatch()`

- **Script naming:**

```
IndexController::viewAction() => index/view.phtml
```

- **`$view` property in controller:**

```
$this->view->item = $items->fetchRow(1);
```

- **In view scripts:**

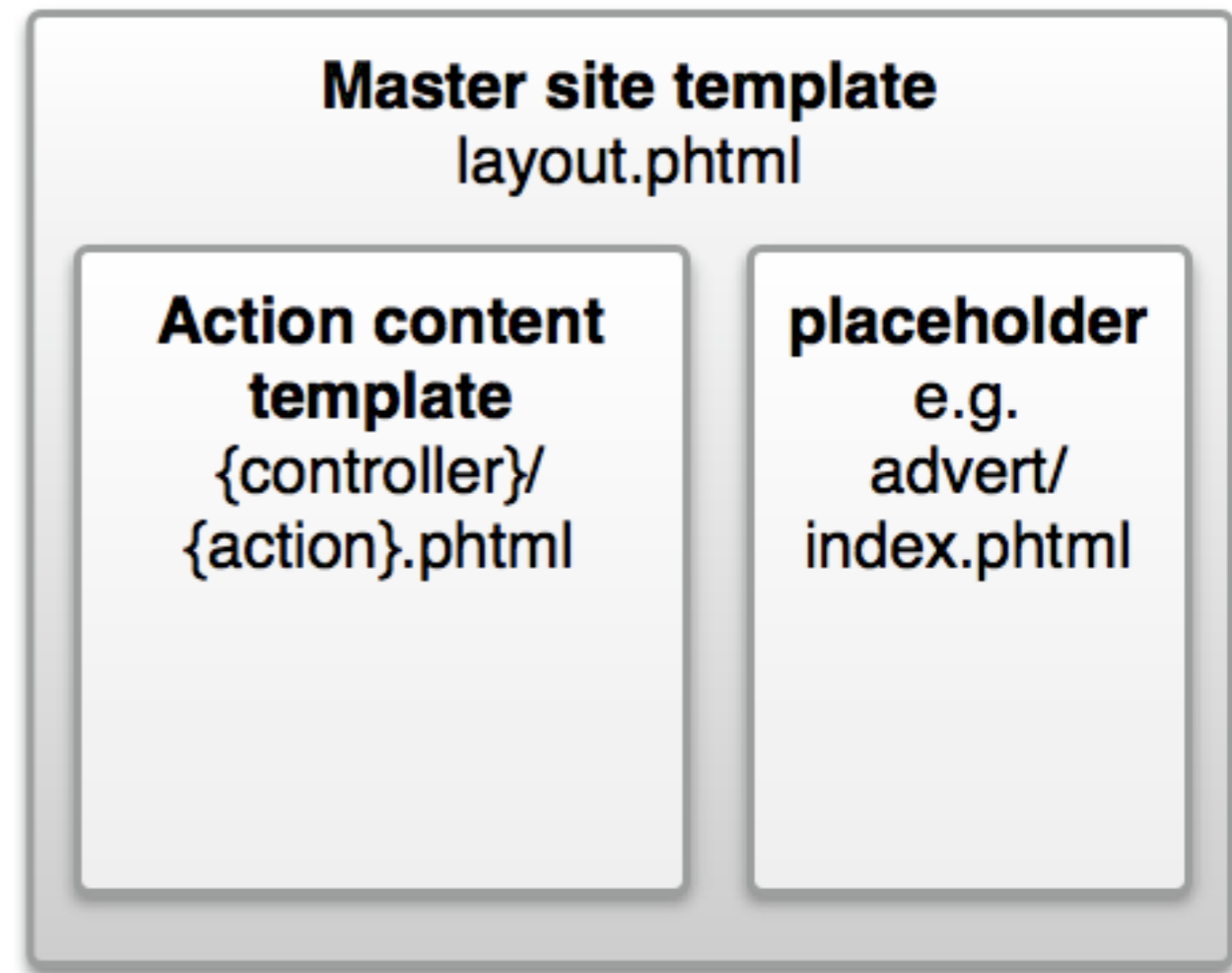
```
<?= $this->escape($this->item->title); ?>
```

View helpers

- Extend view functionality:
 - Format output (e.g. markdown)
 - Encapsulate display logic
- Useful built-in helpers include:
 - `action()`: call a controller action
 - `partial()` / `partialLoop()`: sub templates

Zend_Layout

- Two-Step view
- Render actions to placeholders
- ActionStack plugin to auto-run actions



Action view code

```
<ol>
<?=$this->partialLoop('index/_itemElement.phtml', $this-
>items); ?>
</ol>
</table>

<form action="<?=$this->url(array('action'=>'edit')); ?>"
      method="get">
<p><input type="submit" value="Add New Reason" /></p>
</form>
```

sub-template loops



View helper assembles URL



Layout view code

```
<?= $this->docType('XHTML1_STRICT'); ?>
<head>
<?= $this->headMeta() ?>
<?= $this->headTitle() ?>
<?= $this->headLink() ?>
</head>
<body>
<div id="wrap">
<h1>Is my framework better than yours?</h1>
<h2><?= echo $this->escape($this->title); ?></h2>
<?= $this->layout()->content; ?>
</div>
</body>
</html>
```

Placeholders are
set during dispatch



Action's view output is here



Zend_Db_Table

- OO interface to database table & rows
- Lightweight: Not an ORM.
- Relationships:
 - `findDependentRowset()`
e.g Find all items created by this user
 - `findParentRow()`
e.g Find user who create this item
 - `findManyToManyRowset()`
e.g Find categories of the items created by this user
- Zend_Db database abstraction

Db_Table code

```
class Items extends Zend_Db_Table_Abstract
{
    protected $_name = 'items';
    protected $_rowClass = 'Item';
}
```

← Db table name

```
class Item extends Zend_Db_Table_Row_Abstract
{
    protected function _insert()
    {
        $this->date_created = date('Y-m-d H:i:s');
    }
}
```

← Pre-update hook

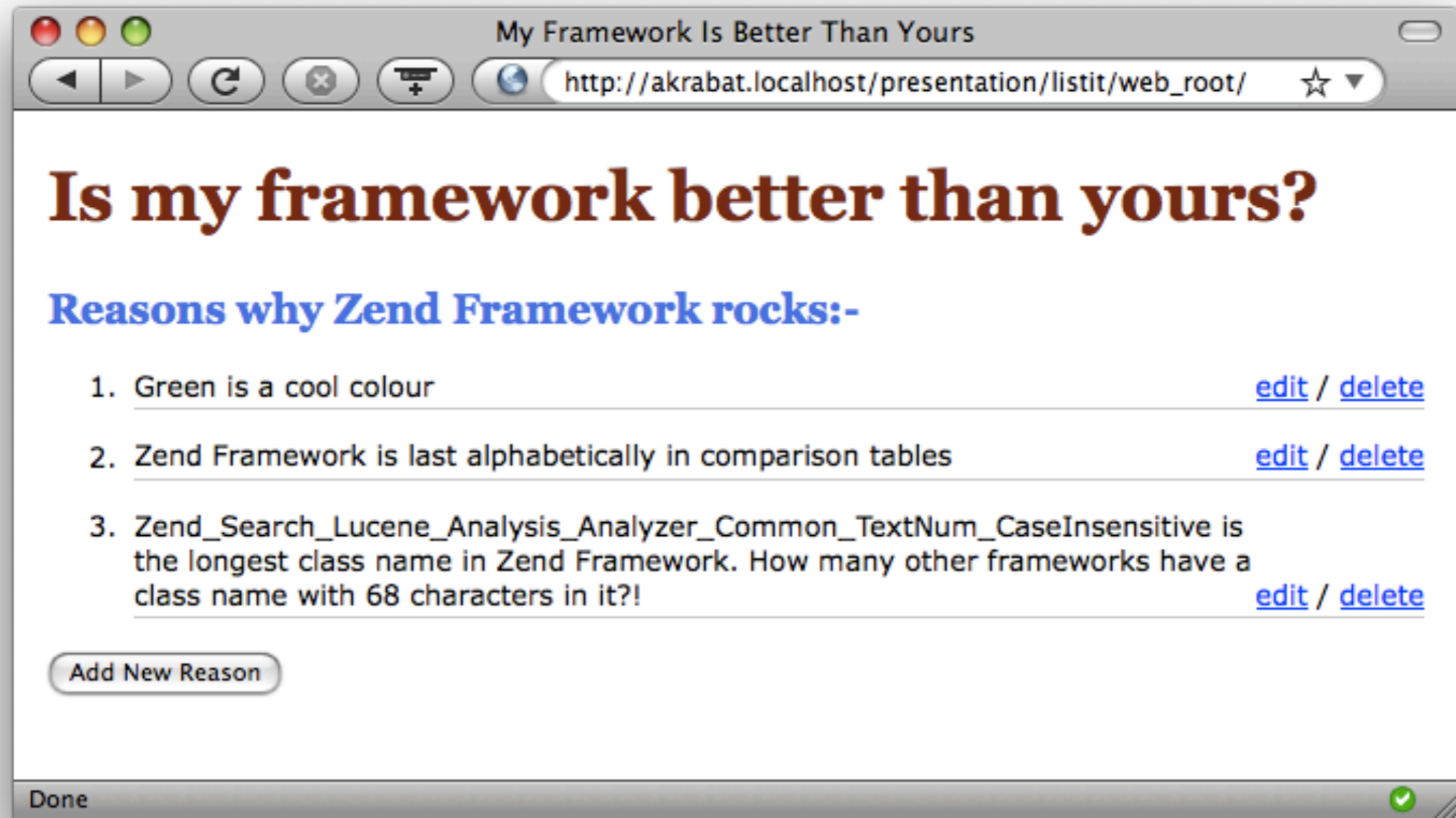
Table usage:

```
$items = new Items();
$recordset = $items->fetchAll();
foreach($recordset as $item) {
    echo $item->reason;
}
```

Row usage:

```
$item = new Item();
$item->reason = 'Stuff';
$item->save();
```

ListIt!



Zend_Form

- Flexible form generation
- Element validation and filtering
- Rendering
 - View helper to render element
 - Decorators for labels and HTML wrappers
- Optional Zend_Config configuration

Building a form in PHP

instantiate element



```
class EditForm extends Zend_Form
{
    public function __construct($options = null)
    {
        parent::__construct($options);

        // setup elements
        $this->addElement('hidden', 'id');

        $this->addElement('text', 'reason');
        $this->reason->size = 50;
        $this->reason->addValidator('NotEmpty');
        $this->reason->addFilter('StringTrim');

        $this->addElement('submit', 'submit');

        // set element decorators
        $this->setElementDecorators(array('ViewHelper'));

        // set form decorators
        $this->addDecorator('FormElements')
            ->addDecorator('HtmlTag', array('tag' => 'p'))
            ->addDecorator('Form');
    }
}
```

add element
to form



Building a form with INI

“id” is element name

```
[form]
elements.id.type = "hidden"

elements.reason.type = "text"
elements.reason.options.size = 50
elements.reason.options.validators.notempty.validator = "NotEmpty"
elements.reason.options.filters.stringtrim.filter = "StringTrim"

elements.submit.type = "submit"
```

```
<?php
$formConfig = new Zend_Config_Ini('edit_form.ini');
$form = new Zend_Form($formConfig->form);
```

Using a form

```
public function editAction()
{
    $items = new Items();
    $id = (int)$this->_getParam('id', 0);
    if ($id > 0) {
        $item = $items->fetchRow('id=' . $id);
    } else {
        $item = $items->createRow();
    }

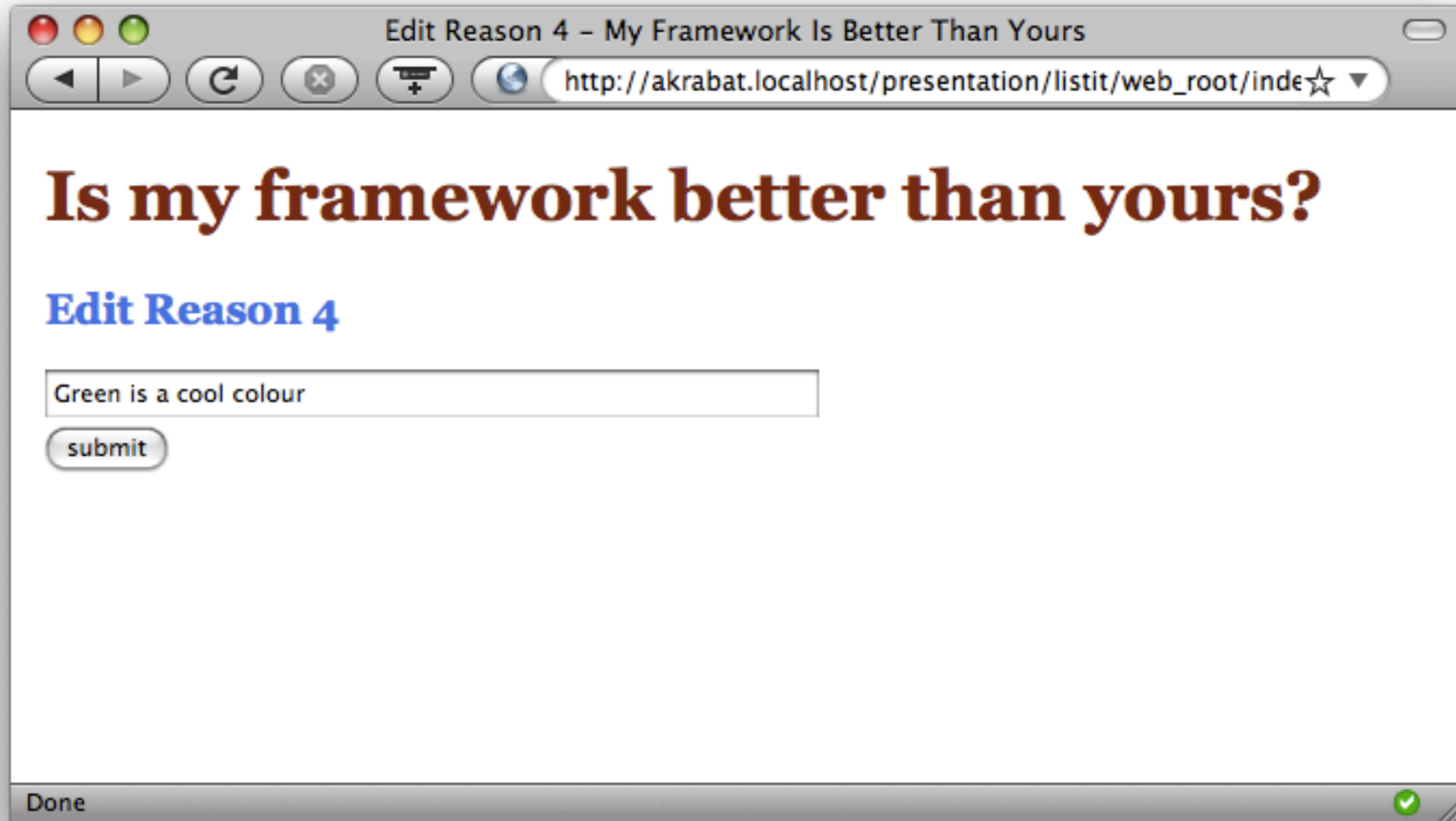
    $form = new EditForm();
    $form->setAction($this->view->url());
    $this->view->form = $form;
    if ($this->getRequest()->isPost()) {
        if ($form->isValid($_POST)) {
            $item->reason = $form->getValue('reason');
            $item->save();
            $this->_gotoHomeAndExit();
        }
    } else {
        $form->populate($item->toArray());
    }
}
```

Model stuff

Process if POSTed

only on first view

Forms in ListIt!



Stuff I haven't mentioned!

- Zend_Search
- Zend_Pdf
- Zend_Mail
- Zend_Cache
- Zend_Auth / Zend_Acl
- Zend_Translate
- Zend_Service
- Zend_Gdata
- Zend_OpenId
- Zend_Date
- Zend_Config
- Zend_Log
- Zend_Feed
- Zend_Json