

# WHAT'S NEW IN ZEND FRAMEWORK 1.6?

By Wil Sinclair, Development Manager  
Matthew Weier O'Phinney, Software Architect  
Alexander Veremyev, Software Engineer  
Ralph Schindler, Software Engineer

# New Features in Zend Framework 1.6

- **Dojo Integration**
- **SOAP**
- **Preview of Tooling Project**
- **Unit Testing Harness for Controllers**
- **Lucene 2.3 Index File Format Support**
- **Zend\_Session save handler for Database Tables**
- **Paginator Component**
- **Text/Figlet Support**
- **ReCaptcha Service**
- **Captcha Form Element**
- **Zend\_Config\_Xml Attribute Support**
- **Zend\_File\_Transfer Component**
- **File Upload Form Element**
- **Zend\_Wildfire Component with FireBug Log Writer**
- **Media View Helpers (Flash, Quicktime, Object, and Page)**
- **Support in Zend\_Translate for INI File Format**

# Dojo Integration



**In May, 2008 Zend announced a collaboration between the Zend Framework and Dojo Toolkit projects. Zend Framework 1.6 contains the first integration points between these 2 projects:**

- **JSON-RPC Server**
- **dojo.data Envelopes**
- **Dojo View Helper**
- **Dijit integration with Zend\_Form & Zend\_View**
- **Dojo Library Distribution**

# SOAP

- **SOAP Server**
- **SOAP Client**
- **Autodiscovery**
- **WSDL Access**
- **WSDL Generation**

# Preview of Tooling Project

**Zend Framework's new tooling system will be shipped as part of the 1.6 release in the 'laboratory' folder. This system is currently under development in the laboratory. For more information:**

**[http://framework.zend.com/wiki/display/ZFPROP/Zend\\_Tool+-+General](http://framework.zend.com/wiki/display/ZFPROP/Zend_Tool+-+General)**

- **Command Line Interface**
- **Project Asset Management**
- **Project Profiles**
- **Fully Extensible Tooling System**

# Unit Testing for Controllers

- **Functional testing for your applications using `Zend_Test_PHPUnit_ControllerTestCase`**
- **Provides assertions for:**
  - HTTP response codes
  - HTTP response headers and redirect locations
  - Last module, controller and/or action processed
  - Response content structure, using CSS selectors
- **Use as part of your quality assurance procedures**
- **Provides all the mock objects you'll need to simulate a request**

# Functional Testing - Example

- Tests showing valid response codes and login

```
public function testIndexActionShouldContainLoginForm()
{
    $this->dispatch('/user');
    $this->assertResponseCode(200);
    $this->assertSelect('form#login');
}

public function testValidLoginShouldInitializeAuthSessionAndRedirectToProfilePage()
{
    $this->request
        ->setMethod('POST')
        ->setPost(array(
            'username' => 'foobar',
            'password' => 'foobar'
        ));
    $this->dispatch('/user/login');
    $this->assertTrue(Zend_Auth::getInstance()->hasIdentity());
    $this->assertRedirectTo('/user/view');
}
```

# Lucene 2.3 Index File Format Support

- **Latest Version of the Lucene Index File Format**

# Session Save Handler for Database

- **Provides support for saving Zend\_Session sessions in a DB table**

# Paginator Component

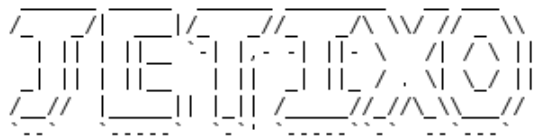
- **Flexible component for displaying large data sets**
- **Can paginate any data, not just data from a relational database**
- **Lazing data fetching**
- **Comes with multiple data source adapters and an interface for implementing additional data source adapters**

# CAPTCHA Support

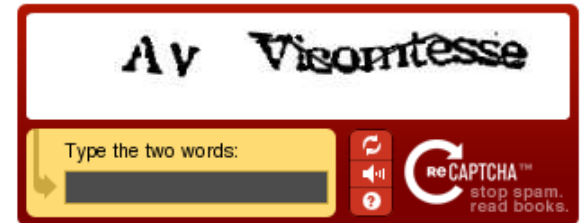
- **Captcha Adapter component with multiple adapters**
  - Figlet – useful for local testing or when GD support or bandwidth are problematic
  - Image – uses GD with Freetype support
  - ReCaptcha – uses Zend\_Service\_ReCaptcha
- **Captcha form element**
  - Choose any captcha adapter
  - Captcha adapter takes care of both validation and decoration

# CAPTCHA - Examples

- Figlet, Image, and ReCaptcha adapters



Figlet CAPTCHA example showing the word "TEST" rendered in a stylized, outlined font.



# Zend\_Config\_Xml Attribute Support

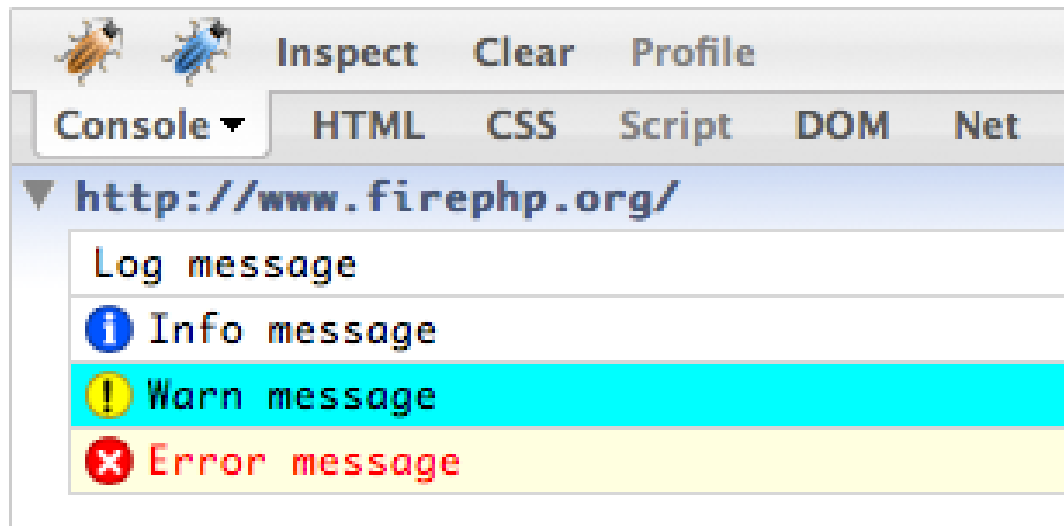
- **Makes Zend\_Config XML documents:**
  - Smaller
  - Human Readable
- **Used for Zend\_Tool's Project Profiles**

# File Upload Form Element

- **Flexible component for displaying data in ‘pages’**
- **Can paginate any data, not just data from a relational database**
- **Lazing data fetching**
- **Comes with multiple data source adapters and an interface for implementing additional data source adapters**

# Wildfire – FireBug Support for ZF

- **Supports Wildfire Protocol:**  
<http://www.wildfirehq.org/>
- **Log to FireBug**
- **Log DB query profiles to FireBug**



# Media View Helpers

- **Flash**
- **QuickTime**
- **Object**
- **Page**

# Support in Zend\_Translate for INI File Format

- **One More Option**
- **Perfect for Name/Value Pairs used Often in Zend\_Translate**

# DOJO INTEGRATION

“

The Zend Framework aims to be a top notch framework for building next generation Web applications in PHP. Dojo has similar aims on the front-end side, so they decided to join forces to provide an integration layer. If you like the combination you will have a better, tighter, way to build your applications.

- ajaxian.com

# Dojo Integration

- **Shipping with Dojo 1.1.1**
- **dojo() view helper for setting up Dojo environment**
  - `dojo.require()` statements
  - `dojo.addOnLoad()` callbacks
  - Layer support
  - And more!
- **Dijit-specific view helpers and form elements and decorators**
- **dojo.data payloads**
- **JSON-RPC server implementation**

# Dojo Integration - Benefits

- **Create rich, cohesive UIs easily and rapidly**
  - Dojo provides a rich set of widgets (Dijits)
  - Dojo makes it simple to extend widgets, as well as to create your own
  - Dijit view and form plugins provide a PHP OOP interface; learn only as much Dojo and/or JavaScript as you need
- **Simplify XHR interactions**
  - ContentPanels can fetch content as needed, on-the-fly
  - Zend\_Dojo\_Data makes feeding various Dojo widgets trivial
- **Get ZF and Dojo in one stop**
- **Create custom Dojo builds when development is done to save bandwidth and increase performance of your Dojo-driven applications**

# Dojo Integration - Examples

- **Dojo-enabled Zend\_Form: sub forms decorated as tabs**

**Sample Form Elements**

Text Elements   Toggle Elements   Select Elements   Slider Elements   Grid Demo

TextBox  
Some Text

PasswordTextBox

DateTextBox  
7/5/2008

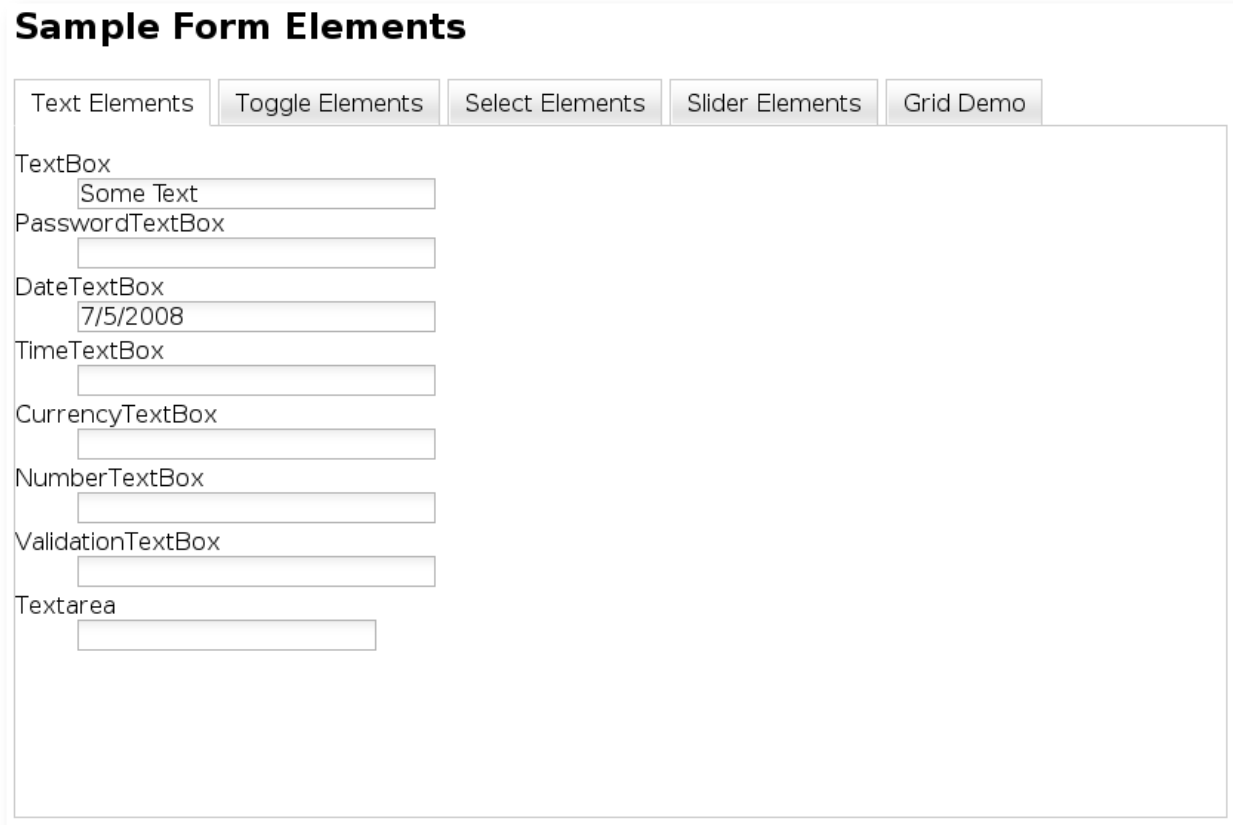
TimeTextBox

CurrencyTextBox

NumberTextBox

ValidationTextBox

Textarea



# Dojo Integration – dojo.data grids

- Use `Zend_Dojo_Data` to return data for `dojox.grid`

## Sample Form Elements

Text Elements   Toggle Elements   Select Elements   Slider Elements   **Grid Demo**

### Example Grid

Class Name	Filename
Zend_Layout_Controller_Action_Helper_Layout	Zend/Layout/Controller/Action/Helper/Layout.php
Zend_Layout_Controller_Plugin_Layout	Zend/Layout/Controller/Plugin/Layout.php
Zend_Layout_Exception	Zend/Layout/Exception.php
Zend_File_Transfer	Zend/File/Transfer.php
Zend_File_Transfer_Adapter_Http	Zend/File/Transfer/Adapter/Http.php
Zend_File_Transfer_Exception	Zend/File/Transfer/Exception.php
Zend_Filter	Zend/Filter.php
Zend_Dojo_Data	Zend/Dojo/Data.php
Zend_Dojo_Form	Zend/Dojo/Form.php
Zend_Dojo_View_Helper_Textarea	Zend/Dojo/View/Helper/Textarea.php
Zend_Dojo_View_Helper_SplitContainer	Zend/Dojo/View/Helper/SplitContainer.php
Zend_Dojo_View_Helper_Dojo_Container	Zend/Dojo/View/Helper/Dojo/Container.php
Zend_Dojo_View_Helper_Form	Zend/Dojo/View/Helper/Form.php

The above grid uses a `dojo.data` store that points to an MVC action to pull data. The data provided is unsorted -- which allows you to sort the data client-side by clicking on a field name.

# Preview of Zend\_Tool

“

Wash, rinse, and repeat. If that feels like the start of every new ZF project, the command line interface for Zend\_Tool will reduce your projects start up time.

- Ralph Schindler ☺

# Preview of Zend\_Tool – Where to Find It

- **Documentation located in the wiki, along with the proposals.**
  - Proposal:  
[http://framework.zend.com/wiki/display/ZFPROP/Zend\\_Tool+-+General](http://framework.zend.com/wiki/display/ZFPROP/Zend_Tool+-+General)
  - Quickstart:  
[http://framework.zend.com/wiki/display/ZFDEV/Zend\\_Tool+-+Quickstart](http://framework.zend.com/wiki/display/ZFDEV/Zend_Tool+-+Quickstart)
- **The development talk is located in a Yahoo Group**
  - <http://tech.groups.yahoo.com/group/zf-tool/>
  - All the latest dev. talk
- **Video demonstration end of week!**

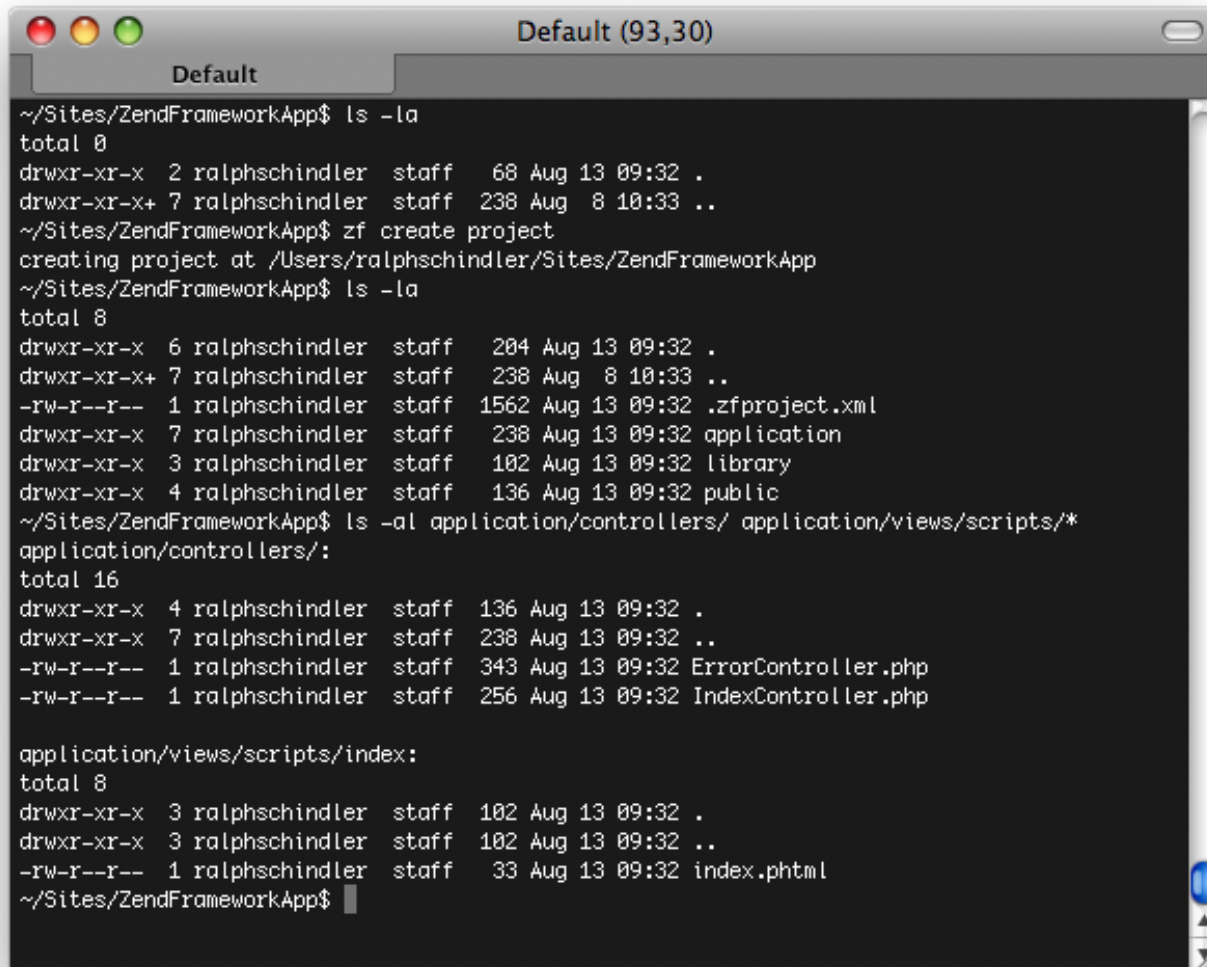
# Preview of Zend\_Tool – What It's Capable of

- **Allows you to create start new projects with one simple command:**
  - Creates project structure
  - Creates public index, bootstrap, and all necessary initial controllers and view
  - Copies ZF into your applications library to create a self contained project
- **Allows you to add new elements to your project with simple commands:**
  - Add actions to controllers and creates respective view scripts
- **You can extent it to add your own functionality**

# Preview of Zend\_Tool – The Breakdown

- **Zend\_Tool\_Rpc**
  - The system responsible for exposing functionality to the developer, most commonly via the command line interface
- **Zend\_Tool\_Project**
  - The system that allows for the creation and management of “project” related elements: directory structure, files, more...
- **Zend\_Tool\_CodeGenerator**
  - The component that allows you to generate PHP code from an OO interface.
- **Zend\_Reflection**
  - Extension of the PHP Reflection system to include docblocks and other PHP elements

# Preview of Zend\_Tool – Just Show Me What It Can Do Already!



```
Default (93,30)
Default
~/Sites/ZendFrameworkApp$ ls -la
total 0
drwxr-xr-x  2 ralpschindler  staff   68 Aug 13 09:32 .
drwxr-xr-x+ 7 ralpschindler  staff 238 Aug  8 10:33 ..
~/Sites/ZendFrameworkApp$ zf create project
creating project at /Users/ralpschindler/Sites/ZendFrameworkApp
~/Sites/ZendFrameworkApp$ ls -la
total 8
drwxr-xr-x  6 ralpschindler  staff  204 Aug 13 09:32 .
drwxr-xr-x+ 7 ralpschindler  staff 238 Aug  8 10:33 ..
-rw-r--r--  1 ralpschindler  staff 1562 Aug 13 09:32 .zfproject.xml
drwxr-xr-x  7 ralpschindler  staff  238 Aug 13 09:32 application
drwxr-xr-x  3 ralpschindler  staff  102 Aug 13 09:32 library
drwxr-xr-x  4 ralpschindler  staff  136 Aug 13 09:32 public
~/Sites/ZendFrameworkApp$ ls -al application/controllers/ application/views/scripts/*
application/controllers/:
total 16
drwxr-xr-x  4 ralpschindler  staff  136 Aug 13 09:32 .
drwxr-xr-x  7 ralpschindler  staff 238 Aug 13 09:32 ..
-rw-r--r--  1 ralpschindler  staff  343 Aug 13 09:32 ErrorController.php
-rw-r--r--  1 ralpschindler  staff  256 Aug 13 09:32 IndexController.php

application/views/scripts/index:
total 8
drwxr-xr-x  3 ralpschindler  staff  102 Aug 13 09:32 .
drwxr-xr-x  3 ralpschindler  staff  102 Aug 13 09:32 ..
-rw-r--r--  1 ralpschindler  staff   33 Aug 13 09:32 index.phtml
~/Sites/ZendFrameworkApp$
```

# Preview of Zend\_Tool – Does It Really “Just Work?”



# SOAP

“

I have some quote to put here. It's a nice one. Not too long but still requires more than 2 lines...

Center your quote to the line

# SOAP – Server

- **Zend\_Soap\_Server class is intended to simplify Web Services server part development.**
- **It's OO wrapper around SOAP PHP extension SoapServer class which gives a flexible way to operate SOAP Server object.**
- **Two operational modes: WSDL and non-WSDL.**

# SOAP – Server

- Common usage:

```
class MyClass {
    /**
     * @param integer $inputParam
     * @return string
     */
    public function method1($inputParam) {
        // ...
    }
    /**
     * @param integer $inputParam1
     * @param string $inputParam2
     * @return float
     */
    public function method2($inputParam1, $inputParam2) {
        // ...
    }
    // ...
}
// ...
$server = new Zend_Soap_Server(null, $options);
$server->setClass('MyClass');
// ...
$server->handle();
```

# SOAP – Client

- **Zend\_Soap\_Client class is intended to simplify Web Services client part development.**
- **Two operational modes: WSDL and non-WSDL.**
- **Base for specific clients development (e.g. Zend\_Soap\_Client\_DotNet is under development now).**

# SOAP – Client

- **Common usage:**

```
$client = new SoapClient("MyService.wsdl");  
//...  
// $result1 is a string  
$result1 = $client->method1(10);  
//...  
// $result2 is a float  
$result2 = $client->method2(22, 'some string');
```

# SOAP – Autodiscovery

- **Generates WSDL description using specified class or set of functions which represent Web Service functionality.**
- **Uses docblock based types definitions.**
- **Generated WSDL could be manually modified, if necessary.**

# SOAP – WSDL Accessor

- **Is utilized by AutoDiscovery functionality to operate with generated WSDL.**
- **Gives a way to operate with WSDL document in terms of Port types, Port operations, Types, Messages, Services and other WSDL entities.**

THANKS! Any Questions?