



Zend Framework: Piece by Piece

ibuildings [i]
THE PHP PROFESSIONALS

Cal Evans
cal@calevans.com

❑ **Glue**

Big ol' box of Legos. All the pieces you need to build something but the structure is up to you.

❑ **Full Stack**

Pre-fabricated structure ready for you to move in and finish. The outer walls are up but you have to build the rooms, lay the flooring, etc.

<http://devzone.zend.com/article/2537-PHP-Abstract-Podcast-Episode-19-Glue-Frameworks-vs.-Full-Stack-Frameworks>

- ❑ Twitter is the new Hello World
- ❑ Stole the idea from <http://www.marketingovercoffee.com/>
- ❑ Goal: Use as many of the listed pull-outs as I could without regard for usefulness of the demo
- ❑ Broken into 3 parts
 - Spider
 - Mines the timeline for like minded people.
 - Cache Runner
 - Displays the cache results in non-destructive way
 - Results Reader
 - Sends email with the results and clears the cache

- Zend_Service_Twitter ← the one we use
- Zend_Service_Amazon
 - EC2
 - S3
 - SQS
 - Others...
- Zend_Service_Yahoo
- Zend_Service_Delicious
- Zend_Service_Flickr
- Etc.

- ❑ Broken into sections
 - Account
 - Status
 - User
 - Direct Messaging
 - Friendship
 - Favorite
- ❑ `getResponseTypes()` & `setResponseType()`
- ❑ OAuth support coming soon (hopefully)

Example

```
$twitter = new Zend_Service_Twitter($username, $password);  
  
$response = $twitter->status->publicTimeline();
```

- My favorite tool in Zend Framework
- Easy to Implement
- Quick results
- Powerful and extensionable

- Core
- Frontend Output
- Frontend Page
- Frontend Function
- Frontend Class
- Frontend File

Zend_Cache Backends

- File
- SQLite
- Memcached
- APC
- Xcache
- ZendPlatform
- TwoLevels
- ZendServer_*
 - Disk
 - Shmem

Example

```
$cache = Zend_Cache::factory('Core',  
    'File',  
    array(  
        'lifetime' => null,  
        'automatic_serialization' => true),  
    array(  
        'cache_dir' =>  
        'd:\\personal\\projects\\pieces.local\\data\\')  
    );  
  
if (!$possibles=$cache->load($cacheName)) {  
    $possibles = array();  
}  
  
$cache->save($possibles,$cacheName);
```

- ❑ Zend_Log
- ❑ Zend_Log_Writer
 - Streams
 - Databases
 - Firebug
 - Email
 - Syslog
 - Mock
- ❑ Zend_Log_Formatter
- ❑ Zend_Log_Filter_Priority

Example

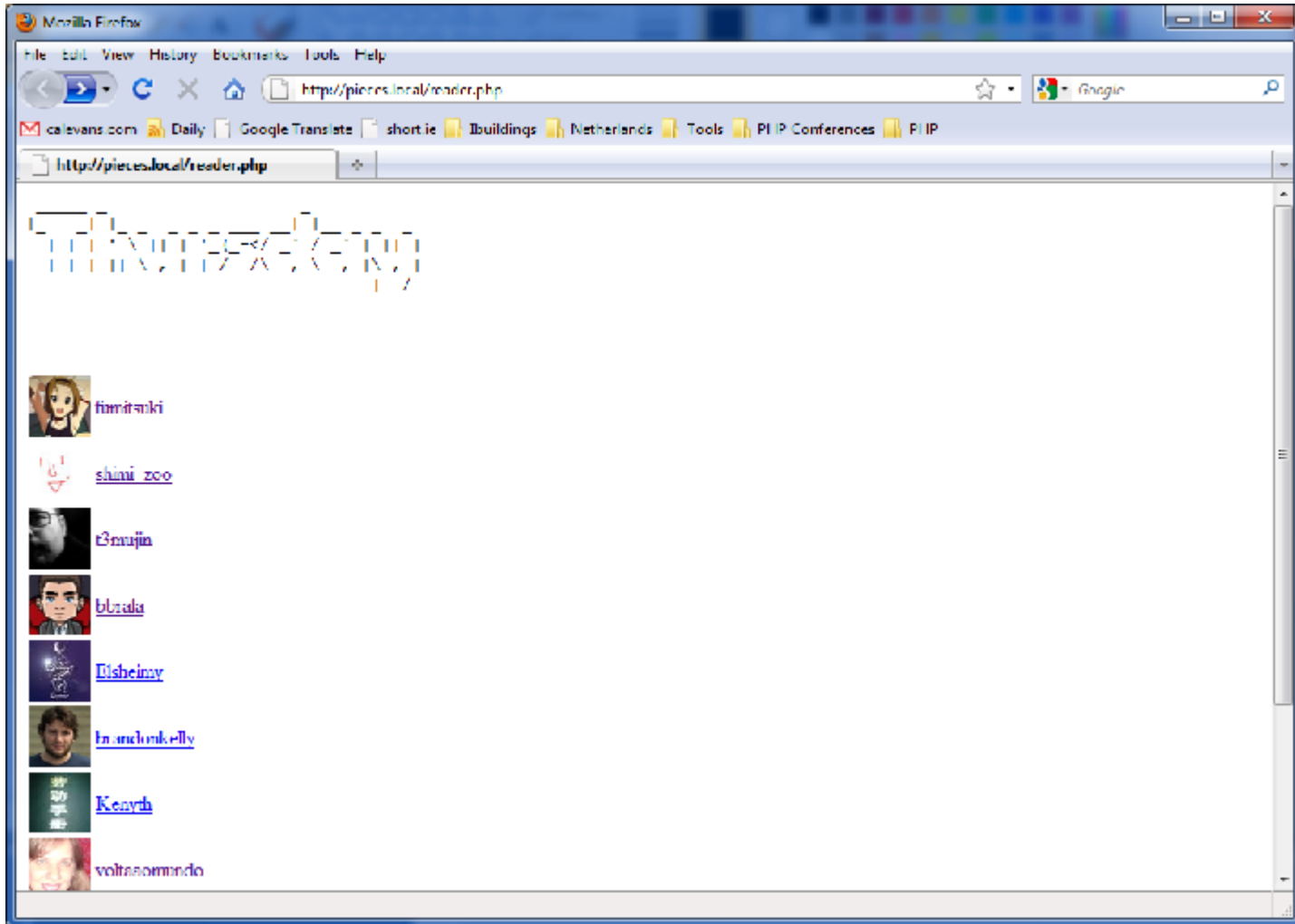
```
$logger = new Zend_Log();  
$writer = new Zend_Log_Writer_Stream('../logs/process.log');  
$logger = new Zend_Log($writer);  
$logger->info('Begin');  
...  
$logger->info('Checking ' . $value->user->screen_name);  
...  
$logger->info('Writing ' . $coolcounter . ' to the cache.');
```

- Great for simple templating needs
- Not just for HTML output
- Must setBasePath()
- Short tags are evil
- Can still use ViewHelpers

Example

```
$view = new Zend_View();
$view->setBasePath('d:\\personal\\projects\\pieces.local\\view');
$view->users = $possibles;
...
$mail->setBodyHtml($view->render('tweeple.php'));
-----
<table>
<?php
foreach($this->users as $id=>$user) {
?>
<tr><td></td><td><a
    href="http://twitter.com/<?PHP echo $user['screen_name'];?>"><?PHP echo
    $user['screen_name'];?></a></td></tr>
<?PHP
}
?>
</table>
```

Example



- ❑ First Zend Component I ever used/ First PHP article I ever wrote
- ❑ Easy to configure
- ❑ Powerful
- ❑ Supports Sending
 - SMTP
 - mail()
- ❑ Supports Reading
 - POP3
 - IMAP
 - MailDir
 - MBox

Example

```
$config = array('auth' => 'login',
                'username' => $mailUser,
                'password' => $mailPasswords,
                'ssl' => 'tls',
                'port' => 587);

$transport = new Zend_Mail_Transport_Smtp('smtp.gmail.com',
    $config);

$mail = new Zend_Mail();
$mail->setBodyHtml($view->render('tweeple.php'));
$mail->setBodyText($view->render('tweeple-plain.php'));
$mail->setFrom('cevens@eicc.com', 'Cal Evans');
$mail->addTo('cal@calevans.com', 'Cal Evans');
$mail->setSubject('People you may want to follow on twitter.');
```

```
$mail->send($transport);
```

- Makes date manipulation easy
- Date formatting
- Date math
- Date Comparisons
- Make sure your Time Zone is set

- ❑ Originally, it would convert data if ext/json was not compiled in. Now, it's just convenient
- ❑ Optional `decode()` parameter allows you to pass in a class name to return. (Default is an associative array)
- ❑ `Encode()` calls `toJson()` if present to allow for encoding of protected/private variables.
- ❑ Translates XML into JSON with `Zend_Json::fromXml()`
- ❑ Not used in the demo but still cool.

```

  _____  _____  _____  _
 |   |   |   | /   |   |   |   |   |
 |   |   |   | |   |   | /   |   |
 |   |   |   | |   |   | /   |   |
 |   |   |   | \   |   | \   |   |
  (Oh common, this one is just fun.)
```

- ❑ Zend Framework implementation of figlet
- ❑ Useful in creating simple captchas
- ❑ Multiple fonts available from <http://www.figlet.org/fontdb.cgi>
- ❑ Fun for simple headers
- ❑ Don't overuse it

- Flesh it out using fully OO and all of Zend Framework
- Archive suggested IDs to eliminate duplicates
- Add spammer detection routine
- Specify language and auto reject any bios that are not in that language.
- `Kateva::makePretty()`

□ Why would you want to do this?

- “Hip Pocket” programs
- Fast prototyping or proof of concept
- Working with another framework and just need a small piece of Zend Framework
- Any time you need a piece of code that already exists in Zend Framework so you don't have to reinvent the wheel.

Who am I?

Cal Evans

Director of the PCE, Ibuildings

<http://techportal.ibuildings.com>

<http://blog.calevans.com>

<http://twitter.com/calevans>

Email: cal@ibuildings.com

Email: cal@calevans.com

AIM: cal@calevans.com

Skype: [cevans-ibuildings](https://www.skype.com/en/contacts/cevans-ibuildings)

IRC: CalEvans

[#phpc](https://www.freenode.net)

