twitter: @akrabat

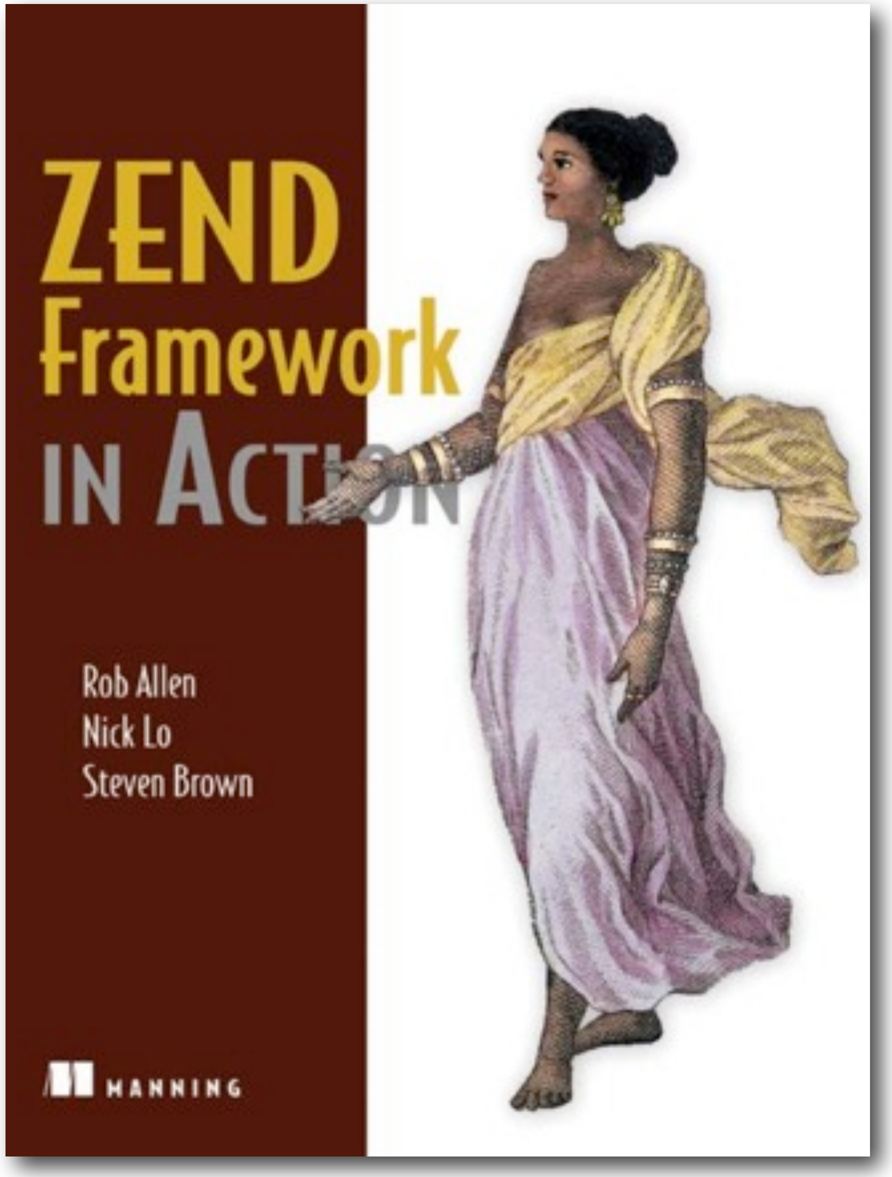
Zend Framework 2.0
What’s new?

Rob Allen

DPC May 2011

Rob Allen?

- PHP developer since 1999
- Wrote Zend_Config
- Tutorial at akrabat.com
- Zend Framework in Action!



Next book!

Zend Framework 2 in Action

(with Ryan Mauger)

Zend Framework 1.0

Some history

- Announced October 2005
- Mar 2006: v0.1 released: not a lot!
- July 2007: v1.0: MVC + cool stuff!
- Mar 2008: v1.5: Form, Layout, Context
- Apr 2009: v1.8: Application, Tool, Nav
- Nov 2010: v1.11: SimpleCloud, UserAgent

Where we are today

- Zend Framework 1 is “done”
- Bug fix mode
- Widespread usage
- Plenty of help available

Zend Framework 2.0

“The primary thrust of ZF 2.0 is to make a more consistent, well-documented product, improving developer productivity and runtime performance.”

Matthew Weier O’Phinney

Think evolution,
not revolution

Focus is on

- Easier learning curve
- Performance
- Easier extension
- Documentation

Zend Framework 2
Key changes

PHP 5.3 support

- Advance the use of PHP 5.3
- Namespaces
- Closures over create_function()

Namespaces

```php
// library/Nennius/Controller/Plugin/InitAdminLayout.php
<?php
namespace Nennius\Controller\Plugin;
use \Zend\Controller\Request\AbstractRequest as Request;
use \Zend\Controller\Plugin\AbstractPlugin as Plugin;

class InitAdminLayout extends Plugin
{
  public function dispatchLoopStartup(Request $request)
  {
    $fc = \Zend\Controller\Front::getInstance();
    $bootstrap = $fc->getParam('bootstrap');
    $layout = $bootstrap->layout;
    $controllerName = $request->getControllerName();
    if (substr($controllerName, 0, 5) == 'admin') {
        $layout->setLayout('admin');
    }
}
```

Interfaces

- Acts as a contract
- Names are adjectives or nouns
  - `Zend\Session\Storage`
- Concrete implementation in sub-namespace
  - `Zend\Session\Storage\ArrayStorage`

Autoloading

1. Standard autoloader
    - Explicit namespace=>path pairs
    - up to 40% performance gain!
2. ClassMap autoloader
    - pre-built array map (for deployment)
    - up to 150% performance gain!

Standard autoloader

```
require_once 'Zend/Loader/StandardAutoloader';
$loader = new Zend\Loader\StandardAutoloader();
$loader->registerNamespace(
    'App', __DIR__ . '/../library/App');
$loader->registerPrefix(
    'PHPExcel_', __DIR__ . '/../library/PHPExcel');
$loader->register();
```

Class-map autoloader

```php
// .classmap.php file
return array(
    array('App\Acl' => __DIR__ . '/Acl.php'),
    array('App\File\Csv' => __DIR__ . '/File/Csv.php'),
);

// within index.php
require_once 'Zend/Loader/ClassMapAutoloader.php';
$loader = new Zend\Loader\ClassMapAutoloader();
$loader->registerAutoloadMap(
    __DIR__ . '/../library/App/.classmap.php'
);
```

Plugin loading

- Alias autoloading
  - map ‘alias’ to class that it represents
  - Interface for consistency (clients)
- Plugin broker (central operations)
- Mapping is simpler to explain!
- Much more performant! (up to 5x!)

Plugin loading

```
$broker = new Zend\View\HelperBroker();
$loader = $broker->getPluginLoader();

$loader->register('url', 'My\View\Helper\Url');

// get a Zend\View\Helper\BaseUrl
$baseUrlHelper = $broker->load('baseUrl');

// get a My\View\Helper\Url
$urlHelper  = $broker->load('url');
```

Exception handling

- Use SPL exceptions for specific catches
- Use an interface for component catches
- Can catch \Exception if you don’t care :)

Exceptions

```
namespace My;
interface ExceptionInterface {}
class Exception implements ExceptionInterface {}
class InvalidArgument extends InvalidArgumentException
   implements ExceptionInterface {}

try {
    $a = new A();
    $a->b($c);
}
catch (InvalidArgument $e) {
    die("Invalid arg happened\n");
}
catch (ExceptionInterface $e) {
    die("Some ‘My’ error happened\n");
}
```

MVC

- Will write to the interface!
- Modules as 1st class citizens
- Page controllers
- Service Locator with DI
- Standardise hooks
- View

Dispatching

- Dispatchable interface
- Anything Dispatchable can be attached to the MVC
- Allows building your own MVC approach

DI & Service Locator

- Service Locator:
  - `set($name, $service)` & `get($name)`
- Dependency Injection container
  - Allows constructor and setter injection
  - Progamattic usage or configuration
  - Used to seed the Service Locator

DI

```php
$db = new Definition('App\Db\Adapter\Sqlite');
$db->setParam('name', 'path/to/file.db');

$users = new Definition('App\Mapper\Users');
$users->addMethodCall('setAdapter',
    array(new Reference('db')));

$di = new DependencyInjector();
$di->setDefinitions(array(
    'db' => $db,
    'users' => $users,
));

$users = $di->get('users');
```

Controllers

```
$userController = new Definition('Site\Controller\User');
$userController->setParam('service',
    new Reference('users'));
$di->setDefition('controller-user', $userController);

// inside dispatcher
$controller = $di->get($controllerName);
$result = $controller->dispatch($request, $response);
```

Zend Framework 2 Development process

git

- Smaller barrier to contribution
- Better branching and merging
- ZF’s git guide: http://bit.ly/zfgit

Unit testing

- Write to /tmp only
- Separate out test asset classes
- Consistent usage of _files directory
- Lower memory/CPU usage

Community Review team

- Smooth the proposal process
- Assist new contributors
- “Fix” orphaned components
- Contact:
  - IRC: #zftalk.dev on freenode
  - Email: cr-team@zend.com

Progress

Milestones

- Autoloading & plugin loading [done]
- Exceptions [done]
- MVC [in progress]
- Testing
- Documentation [proposals in place]
- Internationalisation

When?

Current schedule

- ~~Namespaces milestone: early Aug 2010~~
- ~~Autoloader milestone: early Nov 2010~~
- MVC milestone: end of May 2011
- Beta release: summer 2011
- Stable: by end of year

Migrating from ZF1

Migrating from ZF1

- Namespaces: automatic tool
- Autoloader: simple code change
- MVC: ???

Summary

- Play with the ZF2 snapshots
- Contribute
  - Report bugs!
  - Comment on proposals: http://bit.ly/zf2wiki
- Need help?
  - IRC: #zftalk on freenode
  - Mailing list
- Buy **Zend Framework 2 in Action**

  (when released!)

Questions?

feedback: http://joind.in/3238

email: rob@akrabat.com

twitter: @akrabat

Thank you

feedback: http://joind.in/3238
email: rob@akrabat.com
twitter: @akrabat