

Zend Framework

Jerome Hughes
Consultant
jromeh@gmail.com
630.632.4566

what is Zend Framework?

- a PHP web application framework
- Open Source
- MVC - Model View Controller pattern
- “based on simple, object-oriented best practices”
- <http://framework.zend.com>

Simple & Productive

- Extensible well tested code base
- Flexible architecture
- No config needed to start

Latest web features

- AJAX using JSON
- Search
- Syndication
- Web Services
- High-quality OO PHP5 class library

Licensing

- Friendly & Simple
- Safe for Enterprise use
 - ZF license based on new BSD license
 - Contributor License Agreement based on standard open-source Apache license

Testing baked in

- Unit tested from the start
- Automated unit tests
- TDD - Test driven design
- Enhance with confidence

Components

- ZF is component based
- choose & use only what's needed
- MVC, RAD, DB, Internationalization, Localization, Authentication, Authorization, Session Management, Web Services, Mail, Search, & more!

Getting started with ZF

- great resource... Alan Seiden's "Your First Zend Framework Project on IBM i"
- "From Zero to ZF" on IBM i
- zend.com... Resource/Recorded Webinars/
IBM i Webinars

MVC basics

- Model handles database requests
- View handles user interaction
- Controller mediates between View and Model, determining application flows

why MVC?

- standard pattern & structures simplify complex web application development
- single point of entry with index.php using Front Controller pattern
- standard directory structure implements convention over configuration

History

- started in 2005
- v1.0 July of 2007
- now at 1.11
- developers currently working on 2.0

Installation

- it's installed with Zend Server
- `/usr/local/zendsvr/share/ZendFramework/`
- add to `php.ini` `include_path` variable
- set up SSH link between Zend Studio and IBM i

Create a project

- Create its directory
- Generate skeleton with Zend_Tool
- Configure Apache virtual host
- Test project runs
- Configure and test DB2 access

using ZF on ZS with ZS

- Huh?
- A Zend Framework project runs on Zend Server
- Use Zend Studio (Eclipse like RDP) to edit local version, auto-updates application stored in IBM i IFS

creating directory

- `/www/<appname>/htdocs`
- use 5250 or remote shell in Zend Studio

create project

- use Zend Tool in Zend Studio
- in PHP Explorer, choose Project/Zend Tool
- on zf command line, enter...
 - cd /
 - zf create project zfdemo
 - creates and populates project directories

Link local to remote

- once project is created, link it to IFS
- updates made on client will be automatically saved to host IFS

Configure Apache

- /public is Document Root
- /public/index.php accepts all requests
- ZF router determines controller & action based on request

Apache rewrites

- instruct Apache to send all requests to index.php unless they are present in /public
- afterward, restart Apache
- project runs at `http://<server>:<port>`
- “Welcome to the Zend Framework!”

Controllers & Actions

- use Zend_Tool to add controller
 - cd / zfdemo
 - zf create controller Customer
- create action in controller
 - zf create action list Customer

Configure DB2

- zf configure db-adapter
“adapter=Db2&username=user01&password=secret&dbname=”
- creates application.ini keys for DB2
- add other keys and install i-optimized DB2 adapter, configure so library lists work

access DB2 data

- create model class with Zend_Tool
 - zf create dbtable Custfile CUSTFILE
- edit Custfile.php to tell ZF about table's primary key (expects id)
- now have model that inherits lots of database functionality

add action code

- in controller's `listAction()`
 - create a model object
 - ask the model object to fetch records
 - pass the result set to the view

add code to view script

- .phtml (mixed html & php template)
 - sets up page
 - iterate over result set
 - build table with row for each iteration

data! now what?

- authentication & access control
- form creation and handling
- emails for confirmation & password reset
- links to other pages
- testing!

auth & access control

- Zend_Auth component
- in signup, store salted hashed password
- on login, salt & hash password
- check login password against stored password

activate actions

- AuthController.php activate actions...
 - activateAction()
 - presents activate form, handles issues, generates secret, stores provisional user record, sends confirmation email
 - confirmAction()
 - called by confirmation email link with secret, updates user record to activate

login/logout actions

- AuthController.php login/logout actions...
- loginAction()
 - presents login form, checks credentials, handles issues, stores identity in session
- logoutAction()
 - clears identity in session

reset password actions

- AuthController reset actions...
 - requestresetAction()
 - presents reset form, accepts email, generates secret, updates user record, sends reset email
 - resetpasswordAction()
 - called by reset email link with secret, presents password reset form, updates user record to reset password