View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Research Papers in Economics

INTRODUCTION TO ZEND FRAMEWORK

Dragos-Paul POP
Faculty of Computer Science for Business Management,
Romanian – American University, Bucharest, Romania

ABSTRACT

*A **software framework** provides the skeleton of an application that can be customized by an application developer. Like software libraries, software frameworks aid the software developer by containing source code that solves problems for a given domain and provides a simple API. However, while a code library acts like a servant to other programs, software frameworks reverse the master/servant relationship. This reversal, called inversion of control, expresses the essence of software frameworks.*

*A **web application framework** is a software framework that is designed to support the development of dynamic websites, Web applications and Web services. The framework aims to alleviate the overhead associated with common activities used in Web development. For example, many frameworks provide libraries for database access, templating frameworks and session management, and often promote code reuse.*

1. INTRODUCING ZEND FRAMEWORK

Zend Framework was designed and built to improve developer productivity. Unlike other frameworks that require large configuration files to work, most aspects of a Zend Framework application can be defined at runtime using simple PHP commands. This saves developers time because instead of complex configuration files controlling every aspect of the application, you only configure the parts that deviate from the norm.

The framework was written entirely in PHP 5. It will not run on any server that does not have a minimum of PHP 5.1.4 installed. The current version has been thoroughly tested and over 80% of the code is covered by test cases using PHPUnit.

Zend Framework was built on several key concepts:

- Best Practices
- Community Driven
- Extensionability
- Extreme Simplicity
- Liberal BSD License

Unlike many other frameworks available for PHP, Zend Framework chose not to implement the ActiveRecord pattern and not to ship with an Object-Relation Mapper (ORM). Contrary to popular opinion, this was not an oversight but a conscious decision by the framework team.

The Zend Framework is really a hybrid framework and as such can be used in amuch larger range of projects than strict “application frameworks”. While many components in Zend Framework can be used stand-alone like a component library, it is, at its core an implementation of the “Model-View-Controller” (MVC) pattern.

2. HISTORY AND PHILOSOPHY

Zend Framework was conceived in early 2005 while many new frameworks, such as Ruby on Rails and the Spring Framework, were gaining popularity in the web development community. ZF was publicly announced at the first Zend Conference. At the same time, no widely used framework had been made available to the PHP community to fulfill similar web development needs. The designers of Zend Framework sought to combine the ease-of-use and rapid application development (RAD) features of these

new frameworks with the simplicity, openness, and real-world practicality that is highly valued in the PHP community.

Typically, specific development usage scenarios are implemented using more generalized software components through automatic configuration and/or code generation. In previous releases, the Zend Framework community has opted to complete development and testing of these underlying components before starting work on simplifying development tasks such as database migrations, generating scaffolding, and project creation and configuration. This practice has been the subject of some criticism since some functionality considered by many as necessary for a general release for modern web application frameworks is slated for future Zend Framework releases. Many ZF users, however, have found such generalized software components more reusable and extensible in implementing their applications. Zend Framework also seeks to promote web development best practices in the PHP community; conventions are not as commonly used in ZF as in many other frameworks, rather suggestions are put forth by setting reasonable defaults that can be overridden for each ZF application's specific requirements.

3. THE ZEND FRAMEWORK COMMUNITY

Possibly the greatest asset that Zend Framework has is its community. The community around Zend Framework is growing daily. While several of the developers working on Zend Framework actually work for Zend, the majority of them do not.

The process of proposing and reviewing new components is open and community driven. Because the community is comprised of both beginner and advanced programmers, there is never any shortage of help for new adopters. Whether you prefer mailing lists, forums or chat, Zend Framework community is always there and willing to help. The community realized early on that with any framework, getting started is the hardest part. Therefore, there are numerous tutorials and quickstart guides to help both novice and advanced users get up and running. The project Web site (http://framework.zend.com) houses not only the documentation and downloads for the project but a full bug-tracking/ticketing system that allows anyone to register and submit bugs. This level of openness helps them meet the first goal of the project, "Community Driven".

4. THE ZEND FRAMEWORK LICENSE AND INTELLECTUAL PROPERTY CONCERNS

All contributors to Zend Framework sign a "Contributors License Agreement" stating that the code they are contributing is IP clean. The practice and agreement is similar in nature to that required by the Apache group. This was done not as an exclusionary practice but to give peace of mind to companies considering adopting Zend Framework to build commercial applications. To facilitate its adoption by both open source projects as well as commercial entities, Zend Framework was released under a BSD style license. This allows for the framework to be used in the widest possible range of projects and puts the fewest restrictions on adopters. A complete copy of the BSD License can be found on Zend Framework Web site (http://framework.zend.com/license).

5. MVC

MVC, like so many great things in computers, came out of Xerox's PARC in 1978-1979. These days MVC is a common pattern for frameworks to implement because it separates the code into three logical groups.

- ***The model*** can be thought of as the representation of the data that your application will utilize. In simple terms, the model can be thought of as the "nouns" of your project. An "order", a "member", an "article", these are all examples of potential models in your system.
- ***The view*** contains all the display logic. In the majority of PHP applications, this means the HTML output of your application. However, even in web applications, this can mean a variety of output formats. Whatever the format, the view is responsible for the merging of the data from the model and the actions of the controller and sending it to the proper client. (In most cases with PHP, that's a web browser).

- ***The controller*** is responsible for the domain logic in your applications. It represents the verbs or events. “Add,” “edit” and “submit” are all actions your application can take. The controller embodies these actions for you. There are many good examples of MVC-implemented frameworks in PHP. If that were its only selling point then Zend Framework would be just another framework in an ever-growing list. Zend Framework however, separates itself by allowing you to pull pieces of it out and use them independently. The Zend Framework teams calls this “use at will” architecture. Most of the components that are not part of the MVC core can be pulled out and used as “standalone” components in your application.

Examples of the “use at will” components are

- Zend_Cache
- Zend Rest_Client
- Zend_Feed
- Zend_Log

Each of these can be used independently of the framework itself and that means their functionality can be easily incorporated into existing applications. Simply put, if all you need is a single component, you can use just that component. However when the job requires a full framework, you have that option also.

It should be noted here though that MVC is not something you can retrofit into an existing application. If you are maintaining existing code, the component library aspect of Zend Framework will be of much more interest to you because you can easily integrate the pieces you need without disturbing your existing legacy code. However, if you are building in “green fields” then the MVC aspect of Zend Framework will be of more interest because you have the luxury of building from scratch.

6. FEATURES

- All components are fully object-oriented PHP 5 and are E STRICT compliant
- Use-at-will architecture with loosely coupled components and minimal interdependencies
- Extensible MVC implementation supporting layouts and PHP-based templates by default
- Flexible Table Gateway implementation for accessing data from a relational database in an object-oriented environment
- Support for multiple database systems and vendors, including MySQL, Oracle, IBM DB2, Microsoft SQL Server, PostgreSQL, SQLite, and Informix Dynamic Server
- Authentication and ACL-based authorization using a variety of backend systems
- Data filtering and validation for enhanced application security
- Session management
- Configuration component to promote consistent configuration management throughout the Zend Framework and ZF applications
- Email composition and delivery, retrieval via mbox, Maildir, POP3 and IMAP4
- Indexing and search that supports the Lucene index file format
- Internationalization and localization
- Creation of forms using PHP, configuration files, or XML
- Identity 2.0 technologies such as Microsoft InfoCard and OpenID
- Multiple formats for web services, including XML-RPC, REST, and Google GData.
- Flexible caching sub-system with support for many types of backends, such as memory or a file system.
- Simple logging component inspired by log4j
- Native-PHP component for reading, updating, and creating PDF documents
- Serialization of PHP data structures to and from JSON to facilitate AJAX development
- API for consuming RSS and Atom feeds
- Client libraries for many web services out of the box, including Amazon E-Commerce Service , Akismet, del.icio.us, Flickr, StrikeIron, Yahoo!, Audioscrobbler, and Simpy.

7. ORGANIZATIONS USING ZEND FRAMEWORK:

- brainbits
- Berlin Museums
- Digital Sublimity
- Eurotransplant
- GNU/Linux Matters for Poliglota.
- IBM
- Marseille City School System
- Nokia
- Right Media
- Magento
- Shoppingads
- Australia Week

REFERENCES

[1] Cal Evans, "***php|architect's Guide to Programming with Zend Framework***", Marco Tabini & Associates, Inc. 2008

[2] Wikipedia, http://en.wikipedia.org/wiki /Zend_Framework

[3] Zend, http://framework.zend.com/

[4] IBM, http://www.ibm.com/developerw /opensource/library/os-php-zend1/