Analysis of PHP Frameworks

Presented By:

Jeremy Smereka
Thomas Fung
Mihai Oprescu
Michael Bessette
Mikus Lorence

Introduction

- Over time developers use common techniques for the tasks at hand *(e.g. Database connecting, routing, etc.)*

- A developer may package up all these techniques into one big bundle that is interconnected and release it under a fancy name…

And thus a framework is born!

Why use a framework?

- Provide a starting point for your project.
- They often do things that you will do anyways, such as database connection handling, and they do it well.
- They are often managed by a community
- They give you more job opportunities as many employers look for employees with experience in frameworks.

But there is a problem…

- Frameworks often come with a lot of things that you may not ever use in your project (Feature Bloat)
- Also because many frameworks are community driven, it's hard for developers to make standardized decisions. (older versions of Drupal are bad for this)
- Because everyone and their dog has their own ways of doing things, there are a TON of different frameworks.

Oh and there is A LOT of frameworks.

Adventure PHP	Agavi	Akelos	ATK	CakePHP
CodeIgniter	CoughPHP	evoCore	eZ components	FUSE MVC
FuseBox	Horde Applic.	InterJinn	Jelix	KISSMVC
KohanaPHP	Kolibri	Konstrukt	LightVC	Limb3
Lion	Madeam	Maintainable	OpenBiz	P4A
PHP on TRAX	PHPDevShell	PHOCOA	PhpPeanuts	PHPulse
Pluf	Prado	Qcodo	QCubed	PHP Work
Sapphire	Seagull	SOLAR	Stubbles	Swat
Symfony	Tangra	Tigermouse	Xajax	Xataface
Yii	Zend	ZOOP	Flourish	

So how do you choose the framework for your project?

Source - http://ernieleseberg.com/big-list-of-php-frameworks/

Firstly do you need a framework?

- If your building a blog or smaller site, Content Management Systems/Frameworks, such as Wordpress or Drupal, are great starting points.
- Using existing CMS's over Frameworks give you quite a bit of power and flexibility. Often times, even CMS's are using frameworks under-the-hood (for example: Magento uses Zend Framework)

How do you choose a framework?

- Choosing a framework depends on the problem at hand.
- Often times framework A is essentially the same as framework B in what they do, but how they work internally is different.
- Some things to help choose the framework for a project:

1. Performance
2. Unique Features
3. License
4. How familiar they approach things

We’re here to help!

- We decided to analyze from a high level five popular frameworks.
- For each of the frameworks we looked at the licensing, community, documentation, unique features, and compatibility.

- Because of the high level approach, we were unable to do performance tests, but went with developer feedback on aspects of each framework.

And what did we notice?

- Almost all the frameworks are the same.
- A lot of information is biased because each framework is trying to promote itself.
- Many of the frameworks are based on principals from Ruby on Rails.
- Almost all the frameworks use MVC style programming.
- None-the-less lets dive into each of the five frameworks.

Symfony Framework
MIT Open Source License

- PSR-0 Compliant
- Supports the most databases out of the box
- Appears to be the most complete framework in terms of Documentation, Support, and Features.
- RESTful, Database Connection Pooling
- Great Job Opportunities, as well as certification available.
- Very popular in Europe
- Backed by SensioLabs
- Command Line Interface intergration

CakePHP Framework
MIT Open Source License

- Beginner friendly while still being useful for advanced developers
- RESTful
- Popular for rapid-web development, meaning it’s good for quick prototyping.
- Good Documentation and Support
- Strict conventions compared to PHP in general
- Setup in a short amount of time

Yii Framework
New BSD License & MIT License

- High Performance compared to the other frameworks
- Highly Modular
- Beginner Friendly
- PSR-0 Compliant
- Database Connection Pooling
- Handy Setup Wizard
- Target Audience is for Enterprise and SMB

Zend Framework
New BSD License

ZEND FRAMEWORK

- Many job opportunities
- Certification available
- Targeted mostly for Enterprise
- Most used PHP Framework
- Backbone for CMS's such as Magento
- Upgradability
- Major support from Google, Microsoft, IBM
- Large collection of Components from other developers
- Believes in less predefined constraints, allowing for greater flexability and complexity
- Uses MVC but lacks a predefined model

CodeIgniter
Apache/BSD-style License

- Easy to use/learn
- Spectacular Documentation
- “Faster, lighter and the least like a framework” - Rasmus Lerdorf
- Lightweight (few required core system libraries)
- Extensible – Custom Libraries, helpers, class extensions, system hooks
- Eschew Complexity
- Lacks central leadership, not keeping up with PHP so will most likely die

Closing notes

- There are many frameworks out there to choose from, and this small handful is not the best comparison of frameworks available
- Of course this does give you an idea of what some frameworks are capable of, and what similarities they have

Questions & Comments

If you have any questions or comments we will take them now.