

PHP Frameworks

Analysis

Jeremy Smereka
Thomas Fung
Mihai Oprescu
Michael Bessette
Mikus Lorence

Why are there so many PHP frameworks?

Like many things, PHP frameworks are also best if they fit the project in terms of requirements (time, scalability, modularity, etc.).

Getting used to a framework takes time and patience and unlike most languages, PHP frameworks are abundant.

Because there are many ways of solving problems in PHP, a lot of developers have taken to creating their own framework based on the methods of development they prefer to use. This is one of the reasons why more and more PHP frameworks seem to be released while some also disappear. New frameworks might perform better at certain tasks, but usually, they tend to either have a lack of stability, documentation, support or loyal user base.

Here is a list of PHP frameworks:

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

In order to find the most useful approach that a user needs to take, we introduce you to some PHP frameworks that have been around for a while and are also full of new updated features.

Choosing a PHP framework all comes down to how much flexibility the user is willing to give up based in terms of his ability to develop software.

We will analyze: Symfony, CakePHP, Yii, Zend, CodeIgniter



Analysis of: Symfony 2.2

By Jeremy Smereka

First Release: October 22, 2005

Most Recent Release: March 1, 2013

Licensed under: MIT Open Source License

Compatibility: PHP 5.3.2+ Symfony 2.0, 5.3.3+ Symfony 2.1+

Who uses it?

Many large companies and projects currently use Symfony. Drupal, an open source content management framework, is one major player in the Symfony space. The next iteration of Drupal, Drupal 8, will be utilizing Symfony components as part of Drupal core. This is to provide a standardized core, rather than custom core, for new developers who want to build onto Drupal. Notable websites include: Yahoo! Bookmarks, CFL.ca, TED.com, W3Counter, YouPorn, among many others.

Developer Community?

The developer community is large. On LinkedIn there is approximately 1,400 members of the Symfony Developers group (not the only one), googling Symfony2 brings up 1,160,000 resources, and with the recent release of Symfony 2.2 we see that there was approximately 44 developers working on standard Symfony. Finally, the original creators company, SensioLabs, sponsors Symfony.

Unique Features?

Symfony is a project-based framework. This means that rather than using one version, there are several versions you may use depending on your needs. Symfony also has extremely tight integration with because of their “don’t reinvent the wheel philosophy”. As such, open-source projects such as Doctrine, Monolog, Assetic are integrated into Symfony. It also contains a template engine called Twig, which allows for web designers, rather than developers, to easily maintain the display of each page of a website. It also features a debug toolbar, and a built-in profiler. It was one of the first major frameworks to implement Dependency Injection, meaning you can hot-swap out any component of the framework for something else. For example, you don’t want Doctrine use Propel instead. It has Command Line Integration and a massive library of developer bundles (essentially add-ons). Also Symfony uses YAML for settings throughout. This makes things very easy to read and use for just about anyone. Finally, the developers behind the community actively developed new technologies such as Composer, a dependency manager for PHP, which many other projects and even frameworks use.

Development Philosophy?

The main development philosophy found in Symfony is the use of Model-View-Controller, and they do it very heavily. When you create bundles (from the console), Symfony automatically creates a controller, and default view for them. Models are implemented as entities, documents, or couch-documents depending on your flavor of database; SQL, MongoDB, CouchDB respectively. Since Ruby On Rails, Django, and Spring heavily inspired it, Symfony2 may feel familiar to developers in those areas.

Final Notes

One potential downside to Symfony is that because it is maintained primarily by SensioLabs, they do not support older version. Of course, most companies wouldn't support older versions of software with patches beyond a certain period. This means, that utilizing Symfony would need to be maintained. Since they have now entered a new development process, updating to new versions, which provide security and features, will happen about every 6 months. Even just recently they released version 2.2, with version 2.1 being released only in September of 2012.

More Resources on Symfony 2+

<http://symfony.com/> - Official Symfony Website

<http://fabien.potencier.org/> - Original Creators Blog

<http://getcomposer.org/> - Great Dependency Manager

<https://github.com/symfony/symfony-standard> - Symfony Standard Edition GitHub

Sources of Information

<http://laurent.bachelier.name/2010/07/why-i-wouldnt-use-symfony-for-a-new-project/>

<http://en.wikipedia.org/wiki/Symfony>

<http://en.wikipedia.org/wiki/Drupal>

<http://symfony.com/doc/current/index.html>

<http://fabien.potencier.org/article/65/why-symfony>



Analysis of: CakePHP

By Mihai Oprea

First Release: April 1, 2005

Most Recent Release: January 28th, 2013

Licensed under: MIT Open Source License

Compatibility: PHP4.3.2+ for CakePHP 1.x, PHP 5.2.8+ for CakePHP 2.x

What is it?

CakePHP is a free, open-source (MIT License), rapid development framework for PHP. It enables users to build web applications faster and safer (by following proper conventions). It was released in 2005 and has been somewhat based on the concepts of Ruby on Rails. Like most frameworks, CakePHP helps you avoid reinventing the wheel, making your work more productive.

What is special about CakePHP?

CakePHP is a rapid development framework, meaning that it will offer the user the chance of developing a website / web application in a short amount of time. It also has a good documentation base and support (IRC channel and a Q&A style webpage). It uses an MVC architecture, unlike most PHP frameworks and has a straightforward way of explaining how it works for new users; it is one of the most user-friendly frameworks. The MVC framework helps the user separate the business logic from the front-end logic of a web application but also provides the opportunity to develop innovative web solutions. The features offered by the framework that can be very easily implemented is continually updated by various plug-ins and user contributions. Some features offered by CakePHP are user authorization sessions, pagination and form validation among many others.

How strict is CakePHP?

In terms of strictness, PHP is relatively loose compared to other programming languages. However, CakePHP has conventions that are fairly strict, thus guaranteeing consistency across the web application. This comes at the price of not being the most flexible framework available.

Compatibility of CakePHP?

CakePHP used to support PHP 4 and 5 but versions 2.x do not support PHP 4 anymore. While more support for old versions of PHP might come in useful for older systems, it might create redundant code, making the framework less efficient.

How can I compare frameworks and find out which is best for me?

Comparing the technical features of a PHP framework is the way to start. If you are looking to develop a small web application or a demo that still has an MVC model, CakePHP should be one of the contenders. It is also arguably a solid framework for a large web-app development, but there might be better alternatives out there for that.

A summary of CakePHP's advantages and disadvantages:

- + rapid development framework
- + offers PHP 4 compatibility until the 2.x versions
- + easy Apache/SQL setup; can get it set up in a short amount of time
- + great support (irc channel, Q&A website) and documentation
- + or - strict convention: useful for consistency across the application but less flexibility

References

<http://ernieleseberg.com/so-why-so-many-php-frameworks/>
<http://tv.cakephp.org/>
<http://www.arokiait.com/cake-php-development.htm>
<http://mashable.com/2011/06/23/best-php-frameworks/>
<http://www.phpframeworks.com/>
<http://www.arokiait.com/cake-php-development.htm>

Websites using the CakePHP framework:

<http://www.votigo.com/>
<http://www.arrl.org/>
<http://www.flipkey.com/>
<http://www.handsetdetection.com/>
<http://www.advanced-people-search.com/>
<http://iowa.gov/>
<http://surveymoneymachines.com>
<http://inboxdollars.com>



Analysis of: Yii Framework

By Mikus Lorence

First Release: December 3rd, 2008

Most Recent Release: December 30th, 2013

Licensed under: New BSD License

Compatibility: PHP 5.1+ Yii 1.1, PHP 5.3.0+ Yii 2.0

Introduction

Yii is a free, open source, object oriented, PHP web application framework that supports PHP version 5. It was born out of the effort to fix some of the drawbacks of another framework called the PRADO Framework such as inefficiency, a steep learning curve, and difficult customization. The name “Yii” is an acronym for “Yes, it is!”, and is meant to be a response to any first time users’ questions about the capabilities of this framework. Questions such as “Is it fast?”, “Is it secure?”, and “Is it professional?”. While capable of developing virtually any web application, it is most suited for high-traffic applications such as forums, content management systems, and e-commerce systems, citing its sophisticated caching mechanisms and light weight as the reasons. It is also beginner friendly as it comes with extensive documentation and many tutorials on the framework’s website for users of various levels of expertise.

Incorporated Technologies

PRADO: Yii was started as an improvement of PRADO, many of its aspects are borrowed directly from it. These include the component-based and event-driven programming paradigm, modular application architecture, database abstraction layers, internalization and localization, and many other features and patterns.

Ruby on Rails: Yii borrowed some of the underlying principles of Ruby on Rails such as “convention over configuration” and the “active record pattern”. Convention over configuration is a software design paradigm that seeks to decrease the amount of decisions that developers need to make, increasing efficiency. The active record pattern is an architectural pattern found in software that stores its data in a relational database.

jQuery: is integrated into Yii as the foundational JavaScript framework.

Features

As with most modern framework, Yii adopts the MVC design pattern to allow for clean separation of concerns. Yii also developed its own system called the Yii Data Access Object, based on the PHP Data Objects extension. This allows access to many DBMS’ through a single interface. Yii also supports Object Relational Mapping, which simplifies database

programming by eliminating repetitive SQL statements and representing tables as an instance of a class. It also allows users to use their own database libraries if they so wish. Yii comes with AJAX-enabled widgets, built-in authentication and authorization, and a multi-level caching scheme. The framework deals with several common security concerns. It uses Cross-site Scripting Prevention to prevent other web applications from gathering malicious data from a user through script injection. It uses Cross-site Request Forgery Prevention to prevent malicious websites from allowing the user's web browser to perform unwanted actions on a trusted site. Cookie Attack Prevention is used to guard users from having their sensitive session information stolen through cookies. Yii has many other features such as error handling and logging, skinning and theming mechanisms, unit and functionality testing, and automatic code generation, all of which can make the workflow of a project more streamlined and efficient for the developer. In terms of server support, Yii supports both Apache and NginX server configurations.

Performance

In the tests carried out by the developers of Yii themselves, it seemed to surpass the other 4 frameworks covered in this report by reasonable amounts, even by order of magnitude when "Alternative PHP Cache" was enabled. The reason for such efficient performance is Yii's philosophy of "lazy loading" where the framework does not load a class or instantiate an object until that object is used for the first time, eliminating a lot of time spent preloading every functionality regardless of whether or not the user requested it. Performance benchmarks performed by third parties still placed Yii in the mid to high range among various other PHP frameworks.

Popularity

Yii Framework has started gaining popularity among online communities in the recent years due to its ease of use, high performance, and wide range of projects it is applicable to.

References

<http://en.wikipedia.org/wiki/Yii>
<http://www.yiiframework.com/doc/>
<http://www.yiiframework.com/about/>
<https://github.com/yiisoft/yii>



Analysis of: Zend Framework

By Thomas Fung

First Release: June 30, 2007

Most Recent Release: February 21, 2013

Licensed under: New BSD License

Compatibility: PHP 5.3+ Zend 2.x.x, PHP 5.3 Zend 1.x.x

What is Zend Framework?

Zend is a stable open source PHP framework based on PHP 5. It was founded by the main supporters of PHP and Zend. The framework itself takes on a fully object-oriented approach of programming, thus its components are highly modularized. At the moment, the latest version of the Zend framework is 2.1.3. Zend is one of the most popular PHP frameworks in the industry. The framework is also widely supported by major partners such as IBM, Google, and Microsoft.

Zend's Philosophy

The Zend framework utilizes a MVC design approach. However, compared to other major MVC frameworks where the model is clearly defined, Zend does not have a well-defined model. The reason for this is because Zend believes that every project has a unique model implementation, thus creating an abstract model would not be possible. Zend believes that the developer should be free of predefined constraints, and thus have the power, and flexibility to create more complex implementations.

Along with the minimal predefined constraints, Zend believes that minimal coupling of components is important. Hence why Zend is classified as a glue framework. With highly decoupled components, developers can take advantage of the many individual libraries.

Developer Benefits

Zend is known for its simplicity, because it provides developers a large collection of components. The components can provide up to 80% of the functionality of what developers require, thus reducing development time. Due to extensive testing using PHPUnit, the components provided by Zend are exceptionally stable.

Framework Support

The Zend framework has excellent support for their developers. Zend has a dedicated website for its massive community which provides developer-to-developer support through its online forum. Documentation is also excellent, due to the vast amount of sample

code provided by Zend. Zend also delivers video podcasts, which provide developers tutorials for their framework free of charge.

Future Updates and Upgrades

Zend's reputation for future updates is great as demonstrated by version 1 of the framework. Zend will continue providing updates for the version 1 until at least 2014.

Future upgrades is also great as demonstrated by the version 1 of the framework. At the moment, version 1 of the framework is being phased out. In the case that the developer wants to upgrade their framework to version 2, it is possible to do that. Version 2 of the Zend framework has a migration layer which will allow existing code from the previous version of the framework to run.

References

<http://framework.zend.com/about/>

<http://framework.zend.com/about/faq/>

<http://framework.zend.com/manual/2.1/en/ref/overview.html>

<http://net.tutsplus.com/tutorials/php/10-compelling-reasons-to-use-zend-framework/>

<http://www.alanseiden.com/presentation%20slides/Why-Use-Zend-Framework-on-IBM-i.pdf>

<http://hire-programmers.blogspot.ca/2012/01/advantages-of-zend-framework.html>

<http://www.articlesbase.com/computers-articles/what-is-zend-framework-and-its-advantages-3308896.ht>



Analysis of: CodeIgniter 2.1.3

By Michael Bessette

First Release: February 28, 2006

Most Recent Release: October 8, 2012

Licensed under: Apache/BSD-style open source license

Compatibility: PHP 5.2.4

Introduction

CodeIgniter (CI) is a powerful open source PHP framework designed by Rick Ellis of EllisLab Inc. to be a lightweight toolkit for quickly creating simple, elegant, and full featured web applications. Trivial initial configuration, thorough documentation, and multi-platform compatibility make CI perfect for developers who are new to PHP as well as experts desiring expedited development. A large open source community coupled with highly extensible libraries means many solutions to common web application tasks have already surfaced, and are freely available for use alongside the included features. Applications developed using CI have the capacity to be much faster than those built with other frameworks due to minimal required core system libraries, with additional functionality being loaded dynamically upon request. CI became very popular due to its simplicity and vast wealth of online resources. Some examples of work built with CI are: Bandpages.ie, Sprint Center, ATT Center, Orpheum, TweetRoost, Target Center, Gamestack, Orotoro, and Ukulima.

“We make things. That’s all we do. That’s all we care to do. Most everything else is a distraction from that good and important work.” – EllisLab Manifesto

Features

As per the norm, CI is based on the MVC design pattern, allowing for clear separation of concerns and a level of standardization ideal for an extensible open source system. As of version 2, it supports memcached, file-based, and alternative PHP (APC) caching. It adopts a segment based, rather than query string approach to URL generation that produces search engine friendly results. Although there is no direct ORM support, the abstracted database class uses a modified Active Record database pattern, which can act as an ORM and allows the user to edit contained information with minimal scripting. Currently it supports several common databases including MySQL, Oracle, SQLite, Postgres, and ODBC. AJAX libraries and a complementary AJAX framework, while not included in the base system, are available from the CI GitHub wiki. A simple template engine is included, and its optional use requirement helps avoid unnecessary performance costs.

Issues

Purportedly CI lacks strong central leadership, being largely community supported. As EllisLab is preoccupied with the development of other products, updates to the framework are few and far between, affecting largely insignificant changes. With slow development progress, CI is unlikely to remain a first choice for developers looking to exploit the full extent of PHP features. Initially, CI was structured to support the capabilities of PHP 4. The release of version 2 has dropped PHP 4 support entirely, which flies directly in the face of their supposed emphasis on legacy support. Unit testing of CI is minimal, limiting community participation to only the most experienced. Much of the advanced functionality outside the included libraries was added by the community as an after-thought, and many users complain of a lack of modular design. Modular options like “Modular Extensions” and “Sparks” are available to those dedicated to CI, but they are at the mercy of third party developers and come with their own issues.

Conclusion

For its many supported features, CI is very powerful and has an impeccable track record. Spectacular documentation and online resources, high performance, and simplicity make it an excellent choice for new and experienced developers alike. Highly distributed development is aided by multi-platform support, a standardized MVC base, and easy library extension. CI has been utilized to great effect in the past, and remains a solid framework for applications that do not require the use of features in PHP 5.3 or higher.

References

<http://ellislab.com/codeigniter>
<http://heybigname.com/2012/05/06/why-codeigniter-is-dead/>
http://www.macronimous.com/resources/using_codeigniter_for_PHP_application_development.asp
<https://github.com/EllisLab/CodeIgniter/>
<http://www.linkedin.com/groups/CodeIgniter-Modularity-87251.S.215012268>
<http://codesamplez.com/development/codeigniter-2-features>