



Commentaires sur le choix d'un framework : Une comparaison entre CakePHP et Zend Framework

par Chad Kieffer Guillaume Rossolini ([Tutoriels Web](#))

Date de publication :

Dernière mise à jour :


Est-ce que vous étudiez la possibilité d'utiliser un  **framework** pour votre prochain projet ? Si c'est le cas, cet article est pour vous. J'ai réduit mes choix à CakePHP et Zend Framework, et j'ai documenté ici mes résultats.


Traduction de :  **Notes on Choosing a PHP Framework: A Comparison of CakePHP and the Zend Framework**, par Chad Kieffer.


- I - Introduction
- II - Comparatif
- III - Conclusions et recommandations
 - III-A - CakePHP
 - III-B - Zend Framework
 - III-C - Épilogue
- IV - Liens

I - Introduction

Est-ce que vous étudiez la possibilité d'utiliser un  **framework** pour votre prochain projet ? Si c'est le cas, cet article est pour vous. J'ai réduit mes choix à CakePHP et Zend Framework, et j'ai documenté ici mes résultats.

J'ai commencé à me renseigner sur les frameworks PHP il y a quatre ou cinq ans. À l'époque, il y avait bien peu de choix, surtout par comparaison avec le panel disponible pour les développeurs Java. En suivant le conseil d'un ancien développeur Coldfusion, j'ai adopté Fusebox (le framework Colfusion *de facto*). Fusebox 3 proposait un certain nombre de conventions flexibles pour structurer les applications, organiser le code, nommer les fichiers et contrôler le flux. Cela servait convenablement mon équipe de développement et  **les applications que nous avons construites alors** fonctionnent encore.












J'ai pris du recul par rapport au développement pendant deux ans et, à mon retour, j'ai trouvé des frameworks en abondance. J'ai récemment eu besoin de trouver un framework solide basé sur  **MVC** et, après avoir lu des douzaines d'articles et complété plusieurs tutoriels, je me suis concentré sur CakePHP et Zend Framework.






Avant d'aller plus loin, j'aimerais souligner que mon point de vue est celui de quelqu'un ayant une grande expérience en développement d'applications, mais n'ayant aucun diplôme en informatique ou en technologies de l'information. Je ne peux pas dissenter sur les mérites d'un  **modèle de conception** par rapport à un autre ; en revanche, je peux comprendre le besoin de conventions, les avantages de la modularité et l'efficacité de procédures facilement reproductibles.



Veillez remarquer cependant que symfony, Prado, Code Ignitor et de nombreux autres frameworks semblent tous capables de répondre aux besoins de la plupart des projets. Si vous êtes un grand adorateur d'un autre framework, veuillez ne pas en vanter les mérites ici [*NDT : dans les commentaires du blog anglais*]. En revanche, je m'intéresse à votre expérience dans l'apprentissage d'un framework, particulièrement dans un cadre professionnel.



Voici une comparaison et quelques conclusions après avoir passé un peu de temps avec CakePHP et Zend Framework. Mon échelle de qualité est : mauvais, correct, bon, excellent.

II - Comparatif

	CakePHP	Zend Framework	Commentaires
License	 MIT	 BSD	Chacune de ces licenses est assez flexible. Prenez le temps de considérer comment votre application sera utilisée et distribuée. Il est également important de prendre en considération la compatibilité avec des extensions tierces incluses dans votre application.
Compatibilité PHP	4 et 5	5.1.4 et plus	Je suis curieux de connaître les plans du support de PHP 4 pour CakePHP. Je suppose que la focalisation de ZF sur PHP 5 lui donne des avantages de performances.
Documentation	 Bonne	 Excellente	La documentation de CakePHP est bonne mais l'API de ZF est plus complète.
Communauté	 Google groups, IRC (Freenode),  articles	 Wiki,  listes,  Jabber	Les deux frameworks ont des communautés actives d'utilisateurs. En plus des canaux officiels, il y a des sites communautaires tierces que l'on peut trouver facilement par Google. CakePHP a très bien publicisé son framework, si l'on tient compte de l'absence d'entreprise dont dispose ZF.
Tutoriels / exemples	Excellents	Corrects	 Les fourneaux de CakePHP sont une grande collection d'applications et de codes d'exemples soumis par des utilisateurs.  Les


			<p>articles d'IBM Developer Works pour CakePHP sont excellents. Les tutoriels ZF que j'ai lus sont bons mais nécessitent une expérience approfondie de MVC. Cependant,  les vidéos de débutants ZF sont très utiles. CakePHP propose également  des screencasts.</p>
MVC	Strict	Optionnel	<p>Ces points sont ceux qui différencient le plus les deux frameworks et ils sont probablement les facteurs les plus importants au moment du choix final. CakePHP a des conventions très strictes de nommage et d'organisation du code, tandis que ZF contraint à l'utilisation de ses conventions uniquement si l'on adopte MVC.</p>
Conventions	Strictes	Flexibles	cf. ci-dessus
Configuration	Script PHP	Fichiers XML ou INI	<p>Cette différence ne me semble pas être fondamentale. Si vous êtes un amateur d'outils comme ANT, vous pourriez préférer le choix de ZF pour XML.</p>
Abstraction de BDD	PHP, PEAR, ADOdb	 PHP, PDO	<p>Je peux me tromper, mais il me semble que PDO a les faveurs de la communauté PHP.</p>
Sécurité	 Basé sur ACL	 Basé sur ACL	<p>Les deux frameworks prennent la sécurité très au sérieux et proposent une méthode d'authentification et des autorisations à</p>

			base d'ACL. Tous deux approchent ACL d'une manière similaire et proposent un haut degré de flexibilité dans la construction des ACL.
Manipulation de données	Bon	Excellent	Les deux frameworks proposent d'emblée des mécanismes de validation et de purification des données. ZF propose davantage d'  options de validation , mais  la validation avec CakePHP est facilement extensible <i>via</i> PCRE. CakePHP fournit un unique mécanisme de validation, alors que ZF propose plusieurs filtres pour les données.
Cache	Bon	Excellent	Tous deux proposent un cache des pages à base de fichiers. ZF propose divers backends de cache dont APC, SQLite et, bien entendu, Zend Platform.
Sessions	Excellent	Excellent	Tous deux ont une gestion robuste des sessions.
Logs / débogage	Bon	Excellent	Tous deux ont une méthode de log des applications. IMHO, La classe Zend_Debug donne l'avantage à ZF dans cette catégorie.
Templates	Basé sur PHP	Basé sur PHP	Puisque je viens du design d'interfaces, les templates me sont particulièrement chers. Si vous avez développé des templates pour WordPress ou Drupal, vous vous

			sentirez à l'aise avec CakePHP. ZF demande un peu plus de travail pour faire fonctionner les templates (cf.  Zend_View et  Zend_Layout). Vous pouvez intégrer des méthodes tierces, comme Smarty, dans les deux frameworks.
Helpers	Bon	Excellent	Les deux frameworks fournissent des helpers fondamentaux pour gérer AJAX, les formulaires et les conversions temporelles. CakePHP dispose d'un excellent helper absent dans ZF, mais ZF a un plus grand répertoire de helpers.
JavaScript / AJAX	Bon	Correct	CakePHP intègre Prototype et script.aculo.us tandis que la version actuelle de ZF ne supporte aucun framework JavaScript spécifique. ZF supporte JSON et la version 1.5 ajoute des fonctionnalités AJAX pour les formulaires. J'aimerais que les deux proposent un support plus large de frameworks JavaScript tiers, en particulier JQuery et YUI.
Web Services	Bon	Excellent	Tous deux supportent REST, XML-RPC ainsi que le parcours de flux XML fondamentaux. ZF supporte aussi certains services Web connus tels qu'OpenID,

			Reflection, Akismet, Amazon, Audioscrobbler, Delicious, Flickr, Simpy, Strikelron, Technorati et Yahoo.
Localisation	Bon	Excellent	Tous deux proposent la localisation au travers des moyens traditionnels (i18n, l10n). CakePHP est en train d'ajouter et d'améliorer leur support dans la version 1.2 à venir. ZF semble avoir un meilleur support pour la localisation et la traduction.
Tests unitaires	Oui	Oui	Les deux frameworks proposent cette fonctionnalité que trop d'entre nous ignorent ;) Il me reste à évaluer le support des tests unitaires.

III - Conclusions et recommandations

J'espère utiliser bientôt ces deux frameworks, mais à court terme je construirai certainement avec CakePHP un projet de CMS. J'ai l'intention d'utiliser Zend Framework, ne serait-ce que pour améliorer mes capacités en design d'applications, mes connaissances en  POO et ma compréhension de MVC.

III-A - CakePHP

Si vous n'êtes pas encore familier de MVC, si vous avez besoin de PHP 4 ou si vous avez besoin de puissants outils de génération de code, CakePHP est pour vous. Ses capacités de scaffolding, Bake et ses scripts ACL sont des avantages indéniables lorsque vous avez besoin de construire rapidement une application.

III-B - Zend Framework

Si vous connaissez MVC, si vous avez besoin de plus de contrôle sur le design de votre application ou si vous voulez un support intégré pour certains services Web populaires, alors Zend Framework est pour vous. ZF propose un plus grand degré de flexibilité dans le design d'applications, ce qui s'applique efficacement dans des environnements à lourde charge (scaling).

III-C - Épilogue

Concrètement, les deux frameworks évoluent rapidement à cause de la demande de leurs utilisateurs. Les manques de chacun sont remplis peu à peu, et il semble qu'il ne manque pas de demande de développeurs ayant des compétences dans ces frameworks.

N'hésitez pas à partager votre expérience dans l'apprentissage de n'importe quel framework.

IV - Liens

-  [CakePHP, Wikipedia](#)
-  [Comparaison de Frameworks](#)
-  [Cook Up Web Sites Fast with CakePHP](#)
-  [Zend Framework, Wikipedia](#)

