



Smile
MOTORISTES INTERNET

www.Mcours.com

Site N°1 des Cours et Exercices Email: contact@mcours.com

Livre blanc

Frameworks PHP

Nicolas Richeton – Consultant

Version 1.0

Pour plus d'information : www.smile.fr

Tél : 01 41 40 11 00

Mailto : sd@smile.fr



les frameworks PHP

PREAMBULE

Smile

Fondée en 1991, Smile est une société d'ingénieurs experts dans la mise en œuvre de solutions Internet et intranet. Smile compte 150 collaborateurs.

Le métier de Smile couvre trois grands domaines :

- La conception et la réalisation de sites Internet haut de gamme. Smile a construit quelques uns des plus grands sites du paysage web français, avec des références telles que Cadremploi ou Explorimmo.
- Les applicatifs Intranet, qui utilisent les technologies du web pour répondre à des besoins métier. Ces applications s'appuient sur des bases de données de grande dimension, et incluent plusieurs centaines de pages de transactions. Elles requièrent une approche très industrielle du développement.
- La mise en œuvre et l'intégration de solutions prêtes à l'emploi, dans les domaines de la gestion de contenus, des portails, du commerce électronique, du CRM et du décisionnel.

www.smile.fr



les frameworks PHP

Quelques références de Smile

Intranets - Extranets

- Société Générale - Caisse d'Épargne - Bureau Veritas - Commissariat à l'Energie Atomique
- Visual - Vega Finance - Camif - Lynxial - RATP - AMEC-SPIE - Sonacotra - Faceo - CNRS
- AmecSpie - Château de Versailles - Banque PSA Finance - Groupe Moniteur - CIDJ - CIRAD
- Bureau Veritas - Ministère de l'Environnement - JCDecaux - Ministère du Tourisme
- DIREN PACA - SAS - Institut National de l'Audiovisuel - Cogedim - Ecureuil Gestion
- IRP-Auto - AFNOR - Conseil Régional Ile de France - Verspieren - Zodiac - OSEO - Prolea
- Conseil Général de la Côte d'Or - IPSOS - Bouygues Telecom - Pimkie Diramode
- Prisma Presse - SANEF - INRA - HEC

Internet, Portails et e-Commerce

- cadremploi.fr - chocolat.nestle.fr - creditlyonnais.fr - explorimmo.com - meilleurtaux.com
- cogedim.fr - capem.fr - editions-cigale.com - hotels-exclusive.com - souriau.com - pci.fr
- gdf.fr/presse - dsv-cea.fr - egide.asso.fr - osmoz.com - spie.fr - nec.fr - sogeposte.fr
- metro.fr - stein-heurtey-services.fr - bipm.org - buitoni.fr - aviation-register.com - cci.fr
- schneider-electric.com - calypso.tm.fr - inra.fr - cnil.fr - longchamp.com - aesn.fr
- Dassault Systemes 3ds.com - croix-rouge.fr - worldwatercouncil.org - projectif.fr
- editionsbussiere.com - glamour.com - fratel.org - tiru.fr - faurecia.com - cidil.fr - prolea.fr
- ETS Europe - ecofi.fr - credit-cooperatif.fr - odit-france.fr - pompiersdefrance.org
- watermonitoringaliance.net - bloom.com - meddispar.com - nmmedical.fr - medistore.fr
- Yves Rocher - jcdecaux.com - cg21.fr - Bureau Veritas veristar.com - voyages-sncf.fr
- eurostar.com - AON conseil - OSEO - cea.fr - eaufrance.fr - banquepsafinance.com
- nationalgeographic.fr - idtgv.fr - prismapub.com - Bouygues Construction
- Hachette Filipacchi Media - ELLE.fr - femmeactuelle.fr - AnnoncesJaunes.fr

Applications métier

- Renault - Le Figaro - Sucden - Capri - Libération - Société Générale - Ministère de l'Emploi
- CNOUS - Neopost Industries - ARC - Laboratoires Merck - Egide - Bureau Veritas
- ATEL-Hotels - Exclusive Hotels - Ministère du Tourisme - Groupe Moniteur - Verspieren
- Caisse d'Épargne - AFNOR - Souriau - MTV - Capem - Institut Mutualiste Montsouris
- Dassault Systemes - Gaz de France - CFRT - Zodiac - Croix-Rouge Française

Systèmes documentaires Xml

- Centre d'Information de la Jeunesse (CIDJ) - Pierre Audoin Consultants - EDF R&D



les frameworks PHP

Ce livre blanc

Ce document a pour objectif de présenter et comparer les frameworks PHP les plus répandus.

Comme les autres livres blancs publiés par Smile, cet ouvrage s'efforce de réunir :

- Une approche générale des frameworks PHP, des des concepts et fonctionnalités qu'ils proposent.
- Un recensement des principaux frameworks open source du monde PHP
- Une présentation de ces outils, une évaluation de leurs forces, de leurs limites, de leur maturité, ainsi que quelques exemples simples d'utilisation.

Enfin, un tableau comparatif fait la synthèse des fonctionnalités présentes dans chaque framework.



les frameworks PHP

SOMMAIRE

PREAMBULE	2
SMILE	2
QUELQUES REFERENCES DE SMILE	3
CE LIVRE BLANC	4
SOMMAIRE	5
GENERALITES	6
POURQUOI UTILISER UN FRAMEWORK ?	6
COMPOSANTS OU FRAMEWORKS ?	7
PERFORMANCES	7
VERSION DE PHP	8
CONFIGURATION	9
LES FONCTIONNALITES	11
MODELE MVC	11
EVENEMENTS	12
TEMPLATES	12
CACHE	13
ACCES AUX DONNEES	13
CONVENTIONS	15
GENERATION DE CODE	15
ECHAFAUDAGE	15
GESTION DES DROITS	17
URLS CONVIVIALES	17
AJAX	17
LES BIBLIOTHEQUES DE COMPOSANTS	19
PEAR	19
EZ COMPONENTS	23
LES FRAMEWORKS	27
ZENDFRAMEWORK	27
CAKEPHP	31
SYMFONY	37
PRADO	43
SEAGULL	47
PHPONTRAX	52
CODEIGNITER	56
COPIX	60
JELIX	66
WACT	70
ZOO	71
TABLEAU COMPARATIF	76
CONCLUSION	77



les frameworks PHP

GENERALITES

Pourquoi utiliser un framework ?

Un framework permet de réduire les temps de développement des applications en répondant de façon efficace aux problèmes les plus courants rencontrés par les développeurs. Il inclut généralement de nombreuses fonctionnalités prêtes à l'emploi dont les implémentations sont bien rodées et utilisent des modèles de conceptions standard et bien implémentés. Le temps ainsi gagné sur les questions génériques pourra être mis à profit pour les parties spécifiques de l'application.

De plus, l'utilisation d'un même framework dans différents projets contribue à l'industrialisation du processus de développement : la structure de l'application et les méthodes de programmation sont standardisées, les compétences des développeurs sont réutilisables dans chaque projet.

Un framework est le fruit du travail de dizaines de personnes qui s'appliquent à corriger les problèmes ou les failles de sécurité découverts par l'ensemble des utilisateurs et à proposer de nouvelles fonctionnalités. C'est pourquoi les programmes d'un framework sont en général mieux conçus et mieux codés, mais aussi mieux débogués et donc plus robustes, que ce que pourrait produire un unique programmeur. Outre le gain de temps, on obtient un important gain en termes de qualité.

L'utilisation d'un framework est très fortement recommandée, afin de ne pas avoir à réinventer la roue et d'éviter les erreurs de conception classiques qui créent des problèmes de sécurité ou tout simplement des bugs.



les frameworks PHP

Composants ou frameworks ?

Le terme « *framework* » est très souvent utilisé de façon générique pour désigner deux types d'outils bien différents :

- Les **bibliothèques de composants** qui proposent des modules indépendants, réutilisables et dédiés chacun à une tâche spécifique. Elles permettent d'ajouter facilement des fonctionnalités complexes dans une application, sans constituer un choix structurant vis à vis de l'application.
- Les véritables **frameworks** qui structurent l'application en proposant un ensemble de fonctionnalités standard et **en imposant une architecture logicielle, une arborescence de répertoires et des conventions**. Le framework peut être complété par la suite grâce à un système d'extensions ou en ajoutant de nouvelles fonctionnalités grâce aux bibliothèques de composants.

Ce livre blanc présente successivement des projets appartenant à l'une et à l'autre de ces catégories.

Performances

Beaucoup de développeur pensent que l'utilisation d'un framework ne peut que dégrader les performances d'une application. Ce n'est pas tout à fait exact.

Il est vrai qu'il existe en informatique un compromis très ancien et très habituel entre la qualité de l'architecture logicielle et des outils de développement d'une part, et les performances d'autre part. Avec le temps, ce compromis finit toujours par être tranché en faveur des outils de développement, car la CPU coûte de moins en moins cher, tandis que l'heure de programmeur coûte de plus en plus cher. C'est cette logique qui veut que nous ne développons plus en assembleur, bien que cela puisse être plus performant.

Par cette analyse, on pourrait dire : oui, un développement basé sur un framework sera peut-être moins performant, *mais ça n'est pas grave, car les gains attendus en termes de productivité, de qualité, de robustesse et de maintenabilité sont tellement plus importants au final.*



les frameworks PHP

Mais il faut souligner aussi qu'un framework peut, bien souvent, faire gagner aussi sur le plan des performances. Cela simplement parce qu'il met à disposition du développeur des structures de code plus complexes et plus optimisées que ce qu'il aurait produit seul. L'exemple le plus parlant est bien sûr celui d'un bon gestionnaire de cache, que ce soit sur des objets obtenus d'une base de données, ou bien sur des fragments de page Html : là où un programmeur sans framework aurait eu du mal à réaliser son propre cache de manière complète et fiable, il dispose de classes prêtes à l'emploi qui amélioreront de manière extraordinaire les performances de son application.

Avec PHP, le principal facteur de dégradation des performances vient de ce qu'il faut, à chaque requête, déclarer les différents modules, charger la configuration ou convertir en PHP des templates qui utilisent une syntaxe spécifique.

Les frameworks utilisent plusieurs mécanismes pour minimiser cet impact sur les performances :

- Sélection des modules : le développeur peut définir les modules à utiliser dans une requête en fonction de la page concernée.
- Autoload : utilisation des nouvelles fonctionnalités de PHP5 qui permettent de déclarer des classes ou des modules uniquement au moment de leur utilisation.
- Cache : les fichiers de configuration et les templates non PHP sont traités et convertis dans leur équivalent PHP afin d'être ensuite intégrés de façon beaucoup plus performante lors de chaque requête.

Au final, l'impact est donc faible dans la plupart des cas.

Version de PHP

PHP est disponible en version 5 depuis juillet 2004. Longtemps soucieux de la compatibilité avec la version 4, la plupart des frameworks supportaient à la fois PHP 4 et 5.

Aujourd'hui, de plus en plus de frameworks utilisent des fonctions spécifiques à PHP 5 et ne sont par conséquent plus compatibles avec la version précédente.



les frameworks PHP

PHP4 étant toujours utilisé par de nombreuses entreprises et hébergeurs, la compatibilité avec cette version est dans certains cas un critère de choix important.

Configuration

Le choix du framework va également être dicté par le type d'hébergement. Selon les cas, en particulier si l'hébergement est partagé, il peut être impossible de modifier la configuration de PHP, ni même de disposer des fichiers hors de l'espace dédié au site web. Les points à vérifier sont notamment l'utilisation de Pear et l'activation des fonctions MagicQuotes et ShortTags.

Utilisation de PEAR

Pear¹ est un référentiel de composants PHP, qui sera présenté un peu plus loin dans ce livre blanc. Pear s'installe généralement en dehors de l'espace web du serveur, ce qui pose problème lorsqu'on ne dispose pas d'un accès complet à celui-ci. Il est cependant possible de simuler une installation de Pear en installant les composants nécessaires et en apportant quelques modifications dans le code de l'application.²

MagicQuotes

La fonction Magic Quotes a été introduite dans le langage PHP pour éviter les attaques de type injections SQL. Elle modifie automatiquement les apostrophes et les caractères spéciaux de toutes les informations qui arrivent dans l'application. Cette fonction est activée par défaut pour toute nouvelle installation de PHP.

Elle peut être activée ou désactivée depuis la configuration de php, ce qui complique la tâche des développeurs car elle influence énormément le comportement des applications et les oblige à gérer les différents cas.

Pour une petite application cette fonctionnalité est généralement bénéfique, mais souvent gênantes pour les frameworks qui souhaitent avoir un contrôle total sur les données reçues.

Si certains frameworks sont capables de fonctionner quelle que soit la configuration du serveur, d'autres demandent une configuration

¹ <http://pear.php.net/>

² <http://pear.php.net/manual/en/installation.shared.php>



les frameworks PHP

spécifique. Ces derniers peuvent donc poser des problèmes lorsqu'on n'a pas accès à la configuration de PHP.

Ce problème sera résolu dans PHP6 par la suppression de cette fonction.

ShortTags

Le code php est la plupart du temps délimité par des balises :

```
<?php  
/* code php */  
?>
```

En fonction de la configuration du serveur, il est également possible d'utiliser des balises courtes ou *short tags*.

```
<?  
/* code php */  
?>
```

Cette fonction est désactivée par défaut dans les distributions récentes. Un framework qui utilise les balises courtes posera donc des problème si l'on a pas la possibilité de modifier la configuration de PHP.





les frameworks PHP

LES FONCTIONNALITES

Cette partie présente les fonctionnalités que l'on retrouve le plus souvent dans les frameworks PHP.

Leurs concepteurs ont été fortement influencés par « *Ruby on Rails* », un framework pour le langage Ruby et orienté vers le développement rapide d'applications (RAD). Celui-ci rassemble plusieurs modèles de conception qui ont fait leurs preuves et des règles qui simplifient les développements et limitent les répétitions dans le code de l'application : modèle MVC, ActiveRecord, Scaffolding, ..

On retrouve des fonctionnalités similaires, voir identiques, dans de nombreux frameworks PHP.

Modèle MVC

Le modèle « Modèle Vue Contrôleur » est très répandu dans le développement d'applications et occupe également une place importante dans le développement web. Il permet de structurer une application en distinguant la partie présentation d'une part et le code applicatif d'autre part, ce qui facilite le développement en équipe, la relecture et la maintenance.

Dans le contexte d'une application web, on obtient les éléments suivants :

- **Modèle** : ce sont les données manipulées par le site (les données stockées en base – correspondent au mapping ORM, que l'on abordera plus loin)
- **Vue** : ce sont les différentes pages du site, qui affichent les informations. Le rôle de « vue » est généralement rempli par des templates.
- **Contrôleur** : le traitement des actions utilisateurs. Dans un contexte web, on utilise généralement un contrôleur frontal (Front-Controller) qui reçoit directement les requêtes utilisateur (URL et paramètres) et qui se charge d'exécuter le code adapté : les actions. Les correspondances entre les URL et les différentes



les frameworks PHP

actions sont spécifiées par le développeur et sont souvent appelées « routes » dans les frameworks PHP.

```
// Exemple de route
http://www.monsite.fr/news/list/start/10

-> Controleur : news
-> Action : list
-> Paramètre : start=10
```

Le modèle MVC a fait depuis longtemps ses preuves dans le monde Java avec Struts, Spring, WebWork

Evènements

La programmation événementielle est un modèle différent du modèle MVC qui est surtout pratiqué dans le développement d'applications de type « client-lourd » qui s'exécutent sur le poste de travail de l'utilisateur.

Chaque élément de la page (boutons, liens, champs texte, ... dispose d'évènements correspondant par exemple au clic, à la modification ou à la sélection de l'objet auxquels il est possible d'associer différentes fonctions.

Ce modèle encourage l'utilisation de composants indépendants tels que des formulaires d'authentification, ou des listes d'éléments qui intègre les fonctions de tri et de pagination.

Templates

Les templates (gabarits) rentrent dans le processus de la séparation du code applicatif et de la présentation.

Une template est un fichier contenant essentiellement du code HTML complété de quelques commandes pour l'affichage d'informations issues du code métier.

Il est également possible d'utiliser des templates différents pour l'interface générale du site et pour la mise en page des contenus. Une



les frameworks PHP

template peut inclure d'autres templates, ce qui permet de mutualiser certaines parties des pages.

L'intérêt des templates :

- La présentation est séparée du code applicatif : il est possible de la modifier sans se préoccuper du traitement des données et inversement.
- La création des templates peut être confiée à des graphistes, car elle ne nécessite que très peu de connaissances techniques PHP.
- Il est possible de combiner des templates simples pour former une template plus complexe, et ainsi réutiliser le code au maximum.

Cache

Un système de cache permet de stocker le résultat de l'affichage de certaines pages ou d'actions utilisateur afin de le réutiliser directement lors du prochain accès. La montée en charge des applications s'en trouve améliorée, et les temps d'exécution se rapprochent des pages statiques.

Il est souvent possible de déclarer un cache pour des pages entières ou uniquement pour certaines zones.

Accès aux données

Les applications web accèdent pratiquement à chaque clic à une base de données. L'accès aux données est donc une fonctionnalité essentielle, mais également critique et gourmande en temps de développement lorsqu'on utilise uniquement les fonctions standard de PHP.

Type de base de données

Un premier problème est l'indépendance vis à vis de la base de données. Même si MySQL est la base la plus utilisée dans les applications PHP, on souhaite généralement pouvoir remplacer la base de données lorsque cela est nécessaire. Malheureusement, PHP utilise des API différentes pour chaque base et on constate que l'implémentations du langage SQL



les frameworks PHP

change suffisamment pour empêcher le fonctionnement de l'application avec une base différente que celle avec laquelle elle a été développée.

Un framework peut donc proposer une API unique et des fonctions de création de requêtes SQL, qui génèrent du code SQL adapté à la base de donnée utilisée. La simple modification d'un fichier de configuration permet alors de changer le type de base utilisé.

Mapping Relationnel/Objet et ActiveRecord.

Le framework peut également proposer une fonctionnalité d'ORM (Object Relationnal Mapping), c'est à dire qu'il permet masquer la complexité du langage SQL et d'effectuer la plupart des opérations par l'intermédiaire d'objets très simples, ce qui allège significativement le travail du développeur.

Traditionnellement, un ORM demande un peu de configuration pour créer et enregistrer dans l'application des objets très simples appelés DAO (Data Access Object) qui représentent un élément d'une table donnée. La plupart des implémentations permettent de générer la configuration et les objets directement à partir de la structure de la base elle-même.

L'ORM tel que l'implémente Ruby on Rails sous le nom d'ActiveRecord ne nécessite aucune configuration et inspecte la base de données pour créer les différents objets. Le développement et la maintenance s'en trouvent grandement simplifiés au prix d'une légère perte de performances.

Exemples :

```
// Insertion d'une entrée dans une table
commentaire = new Commentaire();
commentaire->setTitre( « Mon titre »);
commentaire->setCorps( « Mon commentaire »);
commentaire->save();

// Obtenir toutes les entrées d'une table
commentaires = Commentaire.findAll();
```

L'objectif de l'ORM n'est pas de remplacer ou de masquer complètement le langage SQL, mais de prendre en charge la majorité des requêtes utilisées dans l'application. Pour les requêtes les plus complexes ou les plus critiques en vitesse d'exécution, le développeur peut choisir d'utiliser directement le langage SQL si cela est nécessaire.



Conventions

Pendant le développement d'une application, un ensemble de règles sont généralement définies concernant le noms des fichiers ou leur emplacement.

L'utilisation de conventions au niveau du framework permet d'utiliser ces règles pour configurer et lier implicitement les différents modules et classes de l'application, ce qui a pour effet de diminuer le nombre et le volume des fichiers de configuration dans l'application.

Des étapes de configuration sont seulement nécessaires lorsqu'on souhaite s'écarter des réglages par défaut.

La mise en place de conventions prend tout son intérêt lorsqu'elle est couplée à de la **génération de code**.

Génération de code

La mise en place d'un nouveau projet demande généralement la mise en place d'une structure globale et la création de nombreux fichiers.

La génération de code est utilisée pour gagner du temps grâce à l'initialisation automatique de la structure d'une application et à la création et déclaration de nouveaux éléments ou plugins via une simple ligne de commande.

Echafaudage

« L'échafaudage » (ou scaffolding) ajoute sans aucun développement une interface d'administration qui permet l'affichage, l'ajout, la suppression et l'édition d'éléments contenus dans une table de la base de données.

Fortement lié, à la fonctionnalité d'ORM, il permet d'obtenir instantanément une interface temporaire qui pourra plus tard être remplacée par une interface plus évoluée.



les frameworks PHP

En général cette interface est générée à la volée par le framework sans qu'aucun code ne soit présent dans l'application. Il est cependant possible de demander la création du code correspondant et de s'en servir de base pour le développement de la véritable interface.

The NewsEntry has been updated.

List NewsEntries

Id	Title	Body	Created	Modified	Actions
1	Exemple d'interface d'administration	ceci est le corps du message...	2006-08-23 16:06:18	2006-11-29 14:11:08	View Edit Delete
2	Test 2	Deuxième test	2006-11-02 00:00:00	2006-11-29 14:11:21	View Edit Delete

FIGURE 1 – SCAFFOLDING : AFFICHAGE D'UNE TABLE.

Des champs adaptés au types de colonnes : lignes de texte, zones de texte, dates, ou booléens sont utilisés dans les pages d'édition du contenu.

Edit NewsEntry

Title

Body

FIGURE 2 – SCAFFOLDING : EDITION D'UNE ENTREE



les frameworks PHP

Gestion des droits

Le framework peut offrir des méthodes pour définir les rôles des utilisateurs ainsi que les droits nécessaires pour exécuter chaque opération. Il se charge ensuite de vérifier les autorisations à chaque appel d'action et de bloquer l'exécution si nécessaire.

Urls conviviales

Les applications web dynamiques utilisent généralement des paramètres dans les adresses (URL) pour transmettre des informations de page en page.

Les URL obtenues sont difficilement lisibles et ne sont pas très adaptées à l'indexation effectuée par les moteurs de recherches.

Les urls conviviales donnent l'impression à l'utilisateur (ou au moteur d'indexation) d'avoir affaire à un site statique :

```
http://www.monsite.fr/index.php?module=items&action=list
```

devient :

```
http://www.monsite.fr/index.php/items/list
```

et même , si l'on a la possibilité de modifier la configuration du serveur web :

```
http://www.monsite.fr/items/list
```

Ces type d'URL sont souhaitables à la fois pour un bon référencement et pour le confort des visiteurs.

Ajax

AJAX est le nom donné à l'utilisation de Javascript dans un site afin de mettre à jour de façon asynchrone le contenu de certaines parties d'une page.

Un framework prend en charge les fonctionnalités de base de l'application web, ce qui rend difficile de concevoir des requêtes Ajax si le framework n'en prévoit pas l'utilisation.



les frameworks PHP

Le support d'Ajax peut être complété par l'intégration d'une ou plusieurs bibliothèques (prototype, script.aculo.us, YUI, ...), ce qui facilite leur utilisation.

www.smile.fr



LES BIBLIOTHEQUES DE COMPOSANTS

PEAR

Pear³ est un référentiel de composants open source réutilisables pour PHP.

Ces composants utilisent tous les mêmes normes de formatage et de documentation (au format phpDocumentor).



Installation

Pear inclut également un système de distribution et de maintenance utilisable en ligne de commande. Il permet d'installer de nouveaux composants et de mettre à jour les composants déjà installés.

Certains frameworks, même s'ils ne s'appuient pas forcément sur les composants Pear, utilisent tout de même son système de distribution pour gérer l'installation et la mise à jour.

Pear est présent dans chaque distribution de php, un simple script permet de lancer l'installation.

Pear s'installe au même niveau que l'interpréteur php, ce qui rend son installation plus complexe lorsqu'on ne dispose pas d'un accès complet au serveur.

Composants

Les composants Pear sont développés principalement pour PHP4 (mais sont compatibles avec PHP5) et se décomposent en packages dans plusieurs grandes catégories :

³ <http://pear.php.net>



les frameworks PHP

www.Mcours.com
Site N°1 des Cours et Exercices Email: contact@mcours.com

Authentication, Benchmarking, Caching, Configuration, Console, Database, Date and Time, Encryption, Event, File Formats, File System, Gtk, Gtk2, HTML, HTTP, Images, Internationalization, Logging, Mail, Math, Networking, Numbers, Payment, PEAR, PEAR Website, PHP, Processing, Science, Semantic Web, Streams, Structures, System, Text, Tools and Utilities, Validate, Web Services, XML.

Certains composants tels que ceux des packages GTK utilisés pour construire des applications de type client lourd sont plutôt spécifiques, mais on trouvera des composants pour la plupart des fonctionnalités proposées par les frameworks récents :

Pear::Cache_Lite	Système de cache
Pear::LiveUser	Système de gestion des droits utilisateur
Pear::Flexy	Moteur de template
Pear::DB Pear::DB_DataObject	Respectivement pour l'indépendance vis à vis des bases de données, et pour la fonction d'ORM.
Pear::SOAP	Accès aux webservices
Pear::Translation2	Création de sites multilingues.

PEAR inclut de nombreux packages qui ne sont plus maintenus mais sont toujours listés. Un avertissement est présent dans la page d'information des packages concernés.

Exemple Pear:Flexy

Code PHP :

Initialisation de Flexy et transmission des données à afficher :

```
<?php
require_once 'HTML/Template/Flexy.php';
require_once 'PEAR.php';

// Définition des chemins d'accès et de compilation
$options = array(
    'templateDir' => '/var/www/myapp/templates',
```



les frameworks PHP

```

        'compileDir'    => '/var/www/myapp/compiled_templates'
    );

class exemple_template
{
    // Déclaration des données à afficher
    var $comments = array(
        array("id">=1, "title">="Titre 1"),
        array("id">=2, "title">="Titre 2"),
        array("id">=3, "title">="Titre 3")
    );

    function exemple_template ()
    {
        //Initialisation de Flexy
        $output = new HTML_Template_Flexy($options);

        //Compilation de la template
        $output->compile("template.html");

        // Exécution
        $output->outputObject($this);
    }
}

new exemple_template();

?>

```

Template :

```

<h1 class="pageTitle">Commentaires</h1>

<table >
  <tr>
    <th>Id</th><th>Titre</th>
  </tr>
  <tr flexy:foreach="comments,comment">
    <td>{comment[id]}</td><td>{comment[title]}</td>
  </tr>
</table>

```

Documentation

Pear dispose d'un manuel complet, disponible en plusieurs langues et comprenant une documentation de chaque classe ainsi que quelques exemples.

Des articles spécifiques à certains packages sont disponibles sur de nombreux sites Internet.



les frameworks PHP

Licences

Plusieurs licences différentes sont utilisés par les composants Pear : PHP, Apache, LGPL, BSD et MIT.

Elles autorisent toutes la libre utilisation des composants dans toute sorte d'application, mais certaines imposent la redistribution des sources d'un composant en cas de modification (LGPL).

Conclusion

Pear n'est pas un framework, dans le sens où il ne structure pas une application, mais les nombreux composants disponibles permettent de gagner du temps en évitant de réinventer la roue.

Les composants Pear on l'avantage d'être assez connus des développeurs PHP, car cette bibliothèque était très utilisée il y a plusieurs années. On ne basera pas une application uniquement sur ceux-ci.

En général on utilisera ces composants dans un projet PHP4, en complément d'un véritable framework. Il est également possible de les utiliser seuls pour des projets très peu complexes.



eZ Components

eZ Components est un ensemble composants développés et utilisés par la société eZ Systems⁴ pour le CMS Publish.



de
par
eZ

eZ Components nécessite au minimum PHP5.1 et dans certains cas PHP6 pour le support de l'unicode.

cas

L'ensemble est distribué sous license BSD.

Installation

L'installation se fait automatiquement à l'aide de Pear ou bien via le téléchargement d'une archive. Dans ce derniers cas, il sera nécessaire de déclarer le chemin d'accès aux composants au début de chaque script.

L'installation via Pear est conseillée pour une meilleure séparation de eZ Components et du reste de l'application ainsi que pour faciliter les mises à jour.

Même si eZ Components utilise Pear pour son installation, il n'utilise pas de composants de ce dernier.

Composants

Les fonctionnalités couvertes par eZ Components sont à peu près équivalentes à celles de Pear :

- Archive : extraction et compression d'archive Zip et Tar (tar et tar.gz)
- Cache : système de cache capable de stocker un ensemble de valeurs (int, float, string, bool) pendant un temps défini.

⁴ <http://www.ez.no>



les frameworks PHP

- Configuration : lecture/enregistrement de paramètres de configuration.
- ConsoleTools : fonction pour simplifier le développement d'application de type console. Prise en charge des différentes options (-p, -z), Formatage de la sortie(nombre de colonnes, style, etc.).
- Database : permet de générer des requêtes de façon indépendante de la base de donnée. (Pas ORM)
- DatabaseSchema : manipulation de schémas de base de données, application de scripts SQL, comparaison, mise à jour.
- Debug : système de log dédié au debug. Permet de spécifier l'importance de chaque message. Inclus également des timers pour calculer le temps d'exécution de morceaux de code.
- EventLog, EventLogDatabaseTieIn : permettent de stocker des message de debug, alerte, information et audit dans des fichiers où une base de données.
- Execution : permet de capturer les exceptions ou les erreurs fatales PHP et de terminer proprement le script PHP.
- File : fonction de recherche et de suppression récursives.
- Graph : Génération de graphiques
- ImageAnalysis : permet d'extraire les informations Exif(date, taille, etc.) d'une photo.
- ImageConversion : permet de convertir des images tut en effectuant des transformations (changement de taille, application de filtres).
- Mail : envoi (SMTP) et réception (POP3, IMAP) de mails, permet de gérer facilement les pièces jointes.
- PersistentObject : Mapping ORM à partir de définition d'objets.
- PersistentObjectDatabaseSchemaTieIn : Permet de générer automatiquement des définitions d'objets pour



les frameworks PHP

PersistentObject à partir d'un schéma de base de données (cf : DatabaseSchema)

- PhpGenerator : Permet de générer du code PHP.
- SignalSlot : implémente un système de communication entre objets.
- SystemInformation : permet d'obtenir des informations sur le système telles que le système d'exploitation ou la mémoire disponible.
- Template : système de templates.
- Translation : Gestion de l'internationalisation, permet d'obtenir les différentes traductions d'une chaîne.
- TranslationCacheTieIn : Utilisation du cache pour accélérer la gestion des traductions.
- Url : fonctions de manipulation et de décomposition d'urls.
- UserInput : permet de vérifier les informations entrées par l'utilisateur. Champs obligatoires, vérification des données.

Comme dans Pear, l'implémentation du modèle MVC est absente et incombe au développeur.

Template

Exemple d'utilisation du composant ezTemplate :

Appel d'une template :

```
<?php
require_once 'Base/src/base.php';

// Autoload the ezcomponent classes.
function __autoload( $className )
{
    ezcBase::autoload( $className );
}

$t = new ezcTemplate();
echo $t->process( "template.ezt" );
?>
```

Template :



les frameworks PHP

```
<h1>Comments</h1>
<table>
  <tr>
    <th>Id</th>
    <th>Title</th>
  </tr>
  <tr>
    {foreach $comments as $comment}
      <td>{$comment["id"]}</td>
      <td>{$comment["title"]}</td>
    {/foreach}
  </tr>
</table>
```

Documentation

La documentation de eZ Components est très complète. Chaque composant dispose d'un tutorial comprenant généralement une présentation, plusieurs exemples de code et des conseils. Une documentation détaillée de l'API est également disponible.

Conclusion

eZ Components est une bibliothèque, moins connue que Pear, mais activement maintenue. Les différents composants ont déjà fait leurs preuves dans le CMS eZ Publish. Cette bibliothèque est dédiée au PHP5, ce qui la rend plus adaptée que Pear pour les projets récents.

On utilisera généralement eZ Components en complément d'un framework complet, ou bien seuls dans le cas de projet peu complexe.



les frameworks PHP

LES FRAMEWORKS

ZendFramework

Bien qu'il contienne le mot Framework dans son nom, le framework de Zend⁵ est actuellement plus proche de Pear et ezComponents : il regroupe un ensemble de composants.



ou

Malgré tout, il intègre des composants (Controller, RewriteRouter et View) qui permettent de mettre en place une application selon le modèle MVC. La définition de la structure de l'application reste à la charge du développeur.

Le framework Zend est compatible PHP 5.1.4 et distribué sous license BSD.

Composants

Zend contient de nombreux composants. La liste ci-dessous présente les composants principaux, avec en gras les composants utiles pour mettre en place la structure d'un framework :

- **Zend_Cache** : permet de mettre en cache des valeur ou des morceau de code PHP en utilisant différents types de stockage : fichiers, bases de données...
- **Zend_Config** : permet de stocker des informations de configuration dans des tableaux, des fichiers .ini ou XML.
- **Zend_Controller** : base du système MVC, fait le lien entre les urls et différentes actions de l'application.

⁵ <http://framework.zend.com/>



les frameworks PHP

- **Zend_Controller_RewriteRouter** : Permet de gérer les différentes routes de l'application. A utiliser avec Zend_Controller (MVC)
- **Zend_Db** : Couche d'indépendance de la base de données. Permet de construire une application fonctionnant sur plusieurs bases de données.
- **Zend_Feed** : Permet de consommer des flux RSS.
- **Zend_Filter, Zend_Filter_Input** : Permet de filtrer des informations ou chaînes de caractères (suppression de balises interdites, vérification d'adresses email, ..)
- **Zend_Http_Client** : Permet d'effectuer des requêtes GET, POST, PUT ou DELETE vers d'autres sites web.
- **Zend_Json** : fonctions de conversions entre les formats PHP et JSON.
- **Zend_Log** : système de log.
- **Zend_Mail** : Envoi de mails.
- **Zend_Mime** : Manipulation de messages MIME, tels que les emails.
- **Zend_Pdf** : Manipulation de fichiers PDF : création, ouverture, modification et sauvegarde.
- **Zend_Search** : Indexation et recherche full text, basé sur Lucene (Apache).
- **Zend_Service** : Utilisation des webservices Amazon, Flickr et Yahoo.
- **Zend_View** : Système simple de templates à base de fichiers PHP (permet également d'utiliser d'autres moteurs de templates)
- **Zend_XmlRpc** : permet d'exécuter des appels XML-RPC.

D'autres composants sont encore en phase de développement :
 Zend_Acl, Zend_Auth, Zend_Cache_Backend_ZendPlatform,



les frameworks PHP

Zend_Console_Getopt, Zend_Db_Adapter_Oci, Zend_Gdata, Zend_Http_Server, Zend_Date, Zend_Locale, Zend_Measure, Zend_Mail_Folder, Zend_Registry, Zend_Service_Simpay, Zend_Session, Zend_TimeSync.

Les composants Zend_PDF et Zend_Search sont assez uniques. Ce dernier est un portage PHP du projet Lucene de la fondation Apache (Indexation et recherche full text).

Template :

Exemple de template avec le composant Zend_View :

```
Template :
<h1>Comments</h1>

<table>
  <tr>
    <th>Id</th>
    <th>Title</th>
  </tr>

  <?php foreach($this->comments as $key => $comment): ?>
    <tr>
      <td><?php echo $this->escape($val['id']) ?></td>
      <td>
        <?php echo $this->escape($val[title]) ?>
      </td>
    </tr>
  <?php endforeach; ?>
</table>
```

Controlleur :

Le contrôleur frontal est d'abord initialisé :

```
<?php
require_once 'Zend/Controller/Front.php';

Zend_Controller_Front::run('/chemin/des/contrôleurs');
?>

Exemple d'action :
<?php

require_once 'Zend/Controller/Action.php';

class DemoController extends Zend_Controller_Action
{
    public function indexAction()
    {
        //Données
        $data = array(
```



les frameworks PHP



```

        array(
            'id' => '1',
            'title' => 'Entrée 1 - Demo'
        ),
        array(
            'id' => '2',
            'title' => 'Entrée 2 - Smile'
        ),
        array(
            'id' => '3',
            'title' => 'Entrée 3 - Zend Framework'
        )
    );

    //Appel de la template
    Zend::loadClass('Zend_View');
    $view = new Zend_View();
    $view->comments = $data;

    echo $view->render('template.php');
}
?>

```

Documentation

La documentation est très complète et comprend de nombreux exemples pour chaque composant.

Conclusion

Le framework Zend n'est pas un produit fini. Actuellement en version 0.7, chaque nouvelle version voit son lot de nouveaux composants, d'améliorations et de modifications d'API. Il ne dispose donc pas de la maturité que l'on attend d'un framework sur lequel on va baser un projet.

Malgré tout, il sera peut-être à terme la solution la plus utilisée car réalisée par la société Zend qui développe également le langage PHP.

Des frameworks complets basés sur les composants de Zend commencent à apparaître et c'est peut être sous cette forme que s'imposera le Zend Framework.

Dans l'état actuel, on gagnera généralement à choisir un framework plus mature et à utiliser séparément les composants de Zend les plus utiles, tels que Zend_Search et Zend_PDF.



les frameworks PHP

CakePHP

CakePHP⁶ est un framework inspiré de RubyOnRails, compatible avec PHP 4 et 5 et distribué sous licence MIT (open source).

On notera que Cake utilise du code optimisé pour la version de PHP utilisée. Si l'on a le choix, il est conseillé de l'utiliser avec PHP5.



Installation

CakePHP est distribué sous la forme d'une archive contenant le framework et la structure d'un nouveau projet. Une seule action est donc nécessaire pour créer un nouveau projet utilisant CakePHP : décompresser l'archive à l'emplacement souhaité.

Le framework intègre par défaut un fichier de configuration Apache (htaccess) qui active la gestion des urls conviviales. Cependant, si la prise en compte des fragments de configuration est désactivée dans le serveur Web, la modifications de la configuration sera à faire manuellement.

Le framework en lui même ne demande que très peu de configuration. Les seuls impératifs sont les paramètres de connexion à la base de données. Il est également possible d'activer des fonctionnalités de débogage qui permettent d'afficher pour chaque écran les différentes requêtes SQL utilisées ainsi que les temps d'exécution de chacune d'elles.

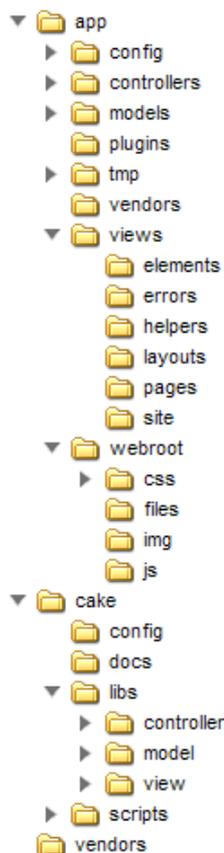
⁶ <http://www.cakephp.org/>



les frameworks PHP

Organisation des répertoires

CakePHP impose une certaine structure de répertoires :



/app contient les fichiers développés pour l'application. Il est possible d'avoir plusieurs dossiers contenant des applications différentes et partageant le même framework.

On y retrouve une séparation de la configuration, des contrôleurs, modèles et vues.

Le dossier **/app/webroot** contient l'espace web accessible : uniquement les images, feuilles de styles, javascripts et fichiers téléchargeables. Tous les autres fichiers se trouvent dans des espace interdisant l'accès direct aux utilisateurs du site.

/cake contient le framework en lui même. Ces fichiers n'ont jamais à être modifiés par les développeurs, ce qui facilite la mise à jour (par simple remplacement du dossier).

/vendors : bibliothèques ou composants supplémentaires, non développées spécifiquement pour le projet et utilisées pour les différentes applications.

Templates

Pour des questions de performances, CakePHP utilise des templates au format PHP. Des templates Smarty peuvent également être utilisées via un plugin dédié.

Template :

```
<h1>Commentaires</h1>
<ul>
  <?php foreach ($comments as $comment):?>
    <li>
      <?php
        echo $html->link($comment['Comment']['title'],
          "/comments/view/".$comment['Comment']['id']);
      ?>
      ( créé le
      <?php echo $comment['Comment']['created']; ?>
      , modifié le
      <?php echo $comment['Comment']['modified']; ?>
    </li>
  </ul>
```



les frameworks PHP

```

)
</li>
<?php endforeach;?>
</ul>

```

Commentaires

- [Entrée 1 - Demo](#) (créé le 2006-08-23 16:40:02 , modifié le 2006-08-23 16:40:02)
- [Entrée 2 - Smile](#) (créé le 2006-08-23 16:41:34 , modifié le 2006-08-23 16:41:34)
- [Entrée 3 - CakePHP](#) (créé le 2006-08-23 16:44:04 , modifié le 2006-08-23 16:44:04)

FIGURE 3 - EXEMPLE CAKEPHP

Génération de code

Cake dispose d'un outil de génération de code en ligne de commande : « Bake ».

Il permet de générer les différents éléments de l'application : Modèles, contrôleurs, vues ...

Exemples :

```

// Générer de la structure d'un nouveau projet
php cake/scripts/bake.php -project nom_projet

// Configuration de l'application (base de données)
php cake/scripts/bake.php -app nom_projet

// Générer un modèle
php cake/scripts/bake.php model nom_modele

//Générer un contrôleur
php cake/scripts/bake.php controller nom_controleur [action1 action2
action3 ... actionN]

// Générer une vue
php cake/scripts/bake.php view nom_controleur/nom_action

```

Une version plus conviviale existe également avec une interface web.

Accès aux données

Cake utilise une implémentation PHP de ActiveRecord. Il est possible, en quelques lignes de créer et mettre à jour les données persistantes. Le



les frameworks PHP

modèle support les relations entre les différents objets. La mise en place se fait via l'héritage de la classe AppModel.

Il est possible d'utiliser des requêtes SQL pour répondre aux cas les plus complexes.

Comme dans Rails, cette implémentation d'ActiveRecord se base sur des conventions pour faire correspondre les objets aux données : le nom des classe détermine les tables à utiliser.

Modèle :

```
class Comment extends AppModel {
    var $name = 'Comment';
}
```

Contrôleur :

```
class CommentsController extends ApplicationController
{
    var $name = 'Comments';

    function index()
    {
        $this->set('comments', $this->Comment->findAll());
    }
}
```

Une interface d'administration peut également être générée à la volée (scaffolding). Dans ce cas, le contrôleur peut être vide.

Contrôleur avec scaffolding :

```
class CommentsController extends ApplicationController
{
    var $name = 'Comments';

    var $scaffold;
}
```

CakePHP permet d'utiliser plusieurs bases de données dans une même application.



les frameworks PHP

Cache

CakePHP dispose d'un système de cache très souple. Il est possible de gérer finement les zones à mettre en cache, grâce à l'utilisation de balises. Le contenu du cache est automatiquement invalidé lors d'une mise à jour des données.

Ajax

CakePHP intègre nativement le support d'AJAX. Toute action est configurée pour répondre soit à une requête HTML classique, soit à une requête AJAX. Des classes de support sont également disponibles.

Sécurité

CakePHP intègre une gestion de la sécurité et permet de définir les droits d'accès de chaque utilisateur aux différents objets et actions.

Aucun mécanisme d'authentification n'est fourni par défaut, mais de nombreux composants additionnels sont disponibles et permettent de supporter différentes méthodes : fichier texte, base de données, annuaire LDAP,...

Extensions

Cake permet l'utilisation de composants, modèles et plugins additionnels afin d'étendre les fonctionnalités. Les composants permettent de rajouter une fonctionnalité, telle que l'authentification. Les modèles permettent d'interagir avec une nouvelle source de données (XML, fichier), et les plugins sont des sortes de mini-applications pouvant comporter un ensemble de vues et d'actions.

Documentation

La documentation est complète, et comporte de nombreux exemples et tutoriaux. CakePHP. La communauté est dynamique et le support est assuré via wiki, mailing list et googlegroups.

L'organisation du framework le rend intuitif et rapide à prendre en main.



les frameworks PHP

Maturité

La première version de CakePHP date de 2005. C'est maintenant un framework mature, disponible en version stable depuis plusieurs mois et qui offre de bonnes performances.

Conclusion

CakePHP est un bon framework, rapide à prendre en main, et qui offre de nombreuses fonctionnalités et de bonnes performances. Il pourra donc être utilisé pour tout nouveau développement spécifique. Il faudra cependant faire attention aux bases de données existantes dont la structure des tables pourrait rentrer en conflit avec les conventions imposées par ActiveRecord.

On regrettera l'absence de fonctions d'internationalisation, mais qui seront présentes dans la version 1.2. Il peut également être complété par des composants issus de Pear ou ezComponents.



Symfony

Symfony⁷ est un framework PHP5 basé sur le modèle MVC et inspiré de Rails. Il est distribué sous licence MIT (open source) et sponsorisé par la société française Sensio.



Installation

Symfony propose une installation automatique à l'aide de Pear, ou bien une installation manuelle à partir d'une archive contenant la structure et la configuration par défaut d'un nouveau projet.

L'installation via Pear est conseillée pour une meilleure séparation entre framework et le reste de l'application ainsi que pour faciliter les mises à jour.

Symfony impose la désactivation de la fonction MagicQuotes dans la configuration de PHP.

A la création du projet, Symfony met en place différents environnements :

- Un environnement de développement, où le cache est entièrement désactivé et les fonctions de débogage et de statistiques sont actives et visibles dans chaque page du projet.

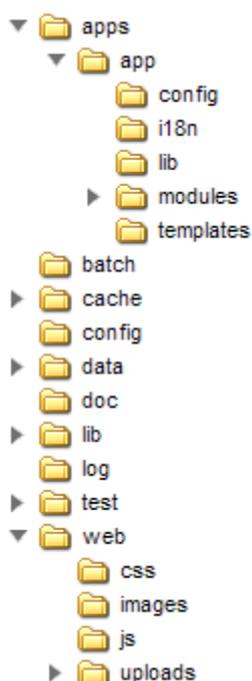
⁷ <http://www.symfony-project.com/>

les frameworks PHP



- Un environnement de production, optimisé pour la vitesse et où le cache est actif.
- Eventuellement un environnement intermédiaire, qui permet de détecter les problèmes pouvant survenir avec les réglages de productions.

Organisation des répertoires :



/apps contient les différentes application (ici une application nommée app). Dans Symfony, les parties front-office et back-office d'une application web sont généralement 2 applications différentes. Il est ainsi possible de n'installer que la partie front office sur des clusters.

Chaque application peut contenir différents modules qui contiennent eux même des actions. Typiquement : MonCMS (application) gère des articles (module) et permet de les lister/afficher/editer (actions).

Si Pear n'est pas utilisé, on retrouve les différentes composantes de Symfony dans les dossiers **/data** et **/lib**.

Le dossier **/web** contient l'espace accessible par l'utilisateur.

Templates

Symfony utilise des templates au format PHP. Il est également possible d'utiliser des templates Smarty bien que le plugin correspondant ne soit pas encore disponible en version stable.



les frameworks PHP

Template :

```
<h1>comments</h1>

<table>
  <thead>
    <tr>
      <th>Id</th>
      <th>Title</th>
      <th>Created</th>
      <th>Modified</th>
    </tr>
  </thead>
  <tbody>
    <?php foreach ($commentss as $comments): ?>
      <tr>
        <td><?php echo link_to($comments->getId(),
'comments/show?id='.$comments->getId()) ?></td>
        <td><?php echo $comments->getTitle() ?></td>
        <td><?php echo $comments->getCreated() ?></td>
        <td><?php echo $comments->getModified() ?></td>
      </tr>
    <?php endforeach; ?>
  </tbody>
</table>
```

comment

Id	Title	Created	Modified
1	Entrée 1 - Demo	2006-08-23 16:40:02	2006-08-23 16:40:02
2	Entrée 2 - Smile	2006-08-23 16:41:34	2006-08-23 16:41:34
3	Entrée 3 - Symfony	2006-08-23 16:44:04	2006-08-23 16:44:04

FIGURE 5 -EXEMPLE SYMFONY

Accès aux données

Symfony utilise l'ORM Propel⁸ pour l'accès aux bases de données.

Propel se base un descriptif de la structure de la base données sous forme XML afin de construire les objets PHP ainsi que pour générer la base de données elle même.

Dans le cas où la base existe déjà il est possible de générer automatiquement le descriptif à partir de la base elle même.

⁸ propel.phpdb.org



les frameworks PHP



Symfony met en place dès le départ une structure d'héritage afin que les développements effectués sur les objets générés ne soit pas écrasés lors des modifications de la structure de la base.

Pour le développeur, le modèle se compose de deux classes pratiquement vides, dans lesquelles il peut rajouter des fonctionnalités spécifiques.

Dans les exemples, Comments représente une entrée dans la base de données, et CommentsPeer l'interface utilisée pour les requêtes et les mises à jour.

Modèle :

Comments.php

```
<?php

// BaseComments est généré par symphony/propel
require_once 'model/om/BaseComments.php';

class Comments extends BaseComments {

}

?>
```

CommentsPeer.php

```
<?php

// include base peer class
require_once 'model/om/BaseCommentsPeer.php';

// include object class
include_once 'model/Comments.php';

class CommentsPeer extends BaseCommentsPeer {

}

?>
```

Controleur :

```
<?php

class commentsActions extends sfActions
{
    public function executeIndex ()
    {
        // Utiliser executeList par défaut.
        return $this->forward('comments', 'list');
    }
}
```



les frameworks PHP

```

    }
    public function executeList ()
    {
        $this->comments = CommentsPeer::doSelect(new Criteria());
    }
}

```

Génération de code

Symfony permet de générer automatiquement la structure des différentes parties du projet :

```

// Créer un nouveau projet
symfony init-project nom_projet

// Créer une nouvelle application
symfony init-app nom_application

// Créer un nouveau module
symfony init-module nom_application nom_module

// Créer le schéma de données à partir d'une base existante
symfony propel-build-schema

// Générer les modèles
symfony propel-build-model

// Scaffolding
symfony propel-generate-crud nom_application nom_module nom_modele

```

Cache

Le système de cache de Symfony est très souple. Il est possible de mettre en cache les actions, les pages, les templates et même des fragments de template.

Le cache peut être activé et désactivé indépendamment dans les différents environnements.

Internationalisation

Les fonctions nécessaires à la création de sites multilingues sont présentes dans Symfony. Les textes et les formats de dates peuvent être automatiquement adaptés à la langue de l'utilisateur.

Ajax

Symfony est parfaitement adapté pour réaliser des sites AJAX : les bibliothèques standards (prototype, script.aculo.us) sont inclus et disposent de classes de support pour faciliter leur utilisation. Il est également



les frameworks PHP

facile de gérer les requêtes AJAX, en créant des actions pour lesquelles le design et la mise en place du site ne sont pas appliqués.

Sécurité

Comme CakePHP, Symfony n'implémente pas de système d'authentification. Il dispose par contre d'un système de gestion de droits qui permet d'interdire l'exécution de certaines actions en fonction des droits de l'utilisateur.

Extensions

Symfony dispose d'un système de plugins. A ce jour plus d'une trentaine d'extensions sont disponibles : génération de flux RSS, de fichiers PDF, système de cache différents...

Documentation

La documentation de Symfony est très complète. De nombreux tutoriaux sont également disponibles.

Maturité

Bien qu'il ne soit pas encore disponible en version finale, Symfony est un framework mature et qui a notamment été sélectionné pour la réalisation de Yahoo! Bookmarks.

Conclusion

Symfony est un bon framework, très complet, bien documenté, et qui dispose d'une communauté active.

Ses performances sont correctes, mais celles de l'ORM utilisé, à savoir Propel, sont critiquées. A terme, Symfony supportera plusieurs ORM et en particulier Doctrine plus récent et plus performant que Propel.

Symfony reste néanmoins l'un des meilleurs frameworks pour les développements spécifiques en PHP.



les frameworks PHP

Prado

Prado est un framework PHP5 qui utilise un modèle événementiel, distribué sous licence BSD.

Il se concentre uniquement sur la partie présentation et n'inclut pas de composant d'accès aux données.

Prado utilise un système de composants et de contrôle qui sont placés sur la page de façon similaire à ASP.NET. Chacun d'eux dispose de méthodes (onLoad, onClick,...) permettant d'exécuter le code correspondant à chaque événement.



Prado permet l'utilisation d'url conviviales via un module dédié.

Installation

Prado est distribué sous la forme d'une archive à extraire dans le répertoire du projet.

Composants graphiques

Prado propose par défaut un grand nombre de composants :

- Les éléments classiques du HTML : Label, bouton, lien, case à cocher, table, iframe, champ texte
- Egalement des éléments plus avancés : sélection de date, de couleur, assistant (navigation à l'aide de boutons « précédent » et « suivant »), zone de texte html filtré, onglets, champ texte riche
- Un ensemble de composants permettant la validation des entrées utilisateurs en utilisant différentes méthodes : champ obligatoire, expression régulière, email, type de données, plages. Ces composants prennent également en charge l'affichage des résultats de validation pour l'utilisateur : messages d'erreurs, utilisation de scripts coté client.



les frameworks PHP

- Enfin, des composants permettent de traiter facilement des ensembles de données : listes, tableaux ...

Bases de données

Prado n'inclut pas de couche d'accès aux bases de données. On pourra utiliser des composants issus de Pear, ezComponents ou ZendFramework pour remplir ce rôle.

Une couche ORM est prévue pour les versions à venir. (3.1)

Templates

Prado dispose d'un système de templates. Contrairement aux autres frameworks, les templates ne servent pas à afficher directement les informations, mais à placer les différents composants dans la page.

Template :

```
<h1>Commentaires</h1>

<com:TForm>
  <com:TDataGrid ID="CommentList"/>
</com:TForm>
```

Contrôleur :

```
<?php

class Home extends TPage
{
  public function onLoad($param)
  {
    $this->CommentList->DataSource= array(
      array('Id'=>'1', 'Nom'=>'Entrée 1 - Demo'),
      array('Id'=>'2', 'Nom'=>'Entrée 2 - Smile'),
      array('Id'=>'3', 'Nom'=>'Entrée 1 - Prado'));
    $this->CommentList->dataBind();
  }
}
?>
```



Commentaires

Id	Nom
1	Entrée 1 - Demo
2	Entrée 2 - Smile
3	Entrée 1 - Prado

FIGURE 6 - EXEMPLE PRADO

Cache

Le cache de Prado permet d'améliorer les performances de l'application en mettant en cache les templates.

Un composants permettent de gérer la mise en cache partielle de certaines parties de la page.

Exemple :

```
<com:TOutputCache>
Contenu mis en cache
</com:TOutputCache>
```

Ajax

Même si le fonctionnement de Prado est parfaitement adapté à l'AJAX, les contrôles utilisant AJAX ne sont prévus que pour la prochaine version (3.1).

De nombreux utilisateurs ont d'ores et déjà commencé à utiliser les versions de développement qui contiennent ces composants (ActiveControl).

Sécurité

Prado propose une API de gestion des droits utilisateurs qui permet de définir des rôles et les autorisations au niveau pages et verbes (méthodes GET et POST utilisées pour l'accès aux pages).

Implémentation par défaut des comptes utilisateurs utilise des fichiers XML et peut être facilement étendue.



les frameworks PHP

Internationalisation

Prado fournit de nombreuses fonctions pour gérer les chaînes de textes et leur traductions.

Documentation

Le guide de démarrage rapide est en réalité plutôt complet. Prado dispose également de document pour les API et d'un forum plutôt actif.

Conclusion

Prado est un bon framework, à compléter avec quelques composants, notamment pour les accès à la base de données.

A cause de son modèle événementiel, il sera surtout conseillé pour les applications web très interactives, plutôt que pour les sites web de consultation.



les frameworks PHP

Seagull

Seagull est un framework PHP4 et 5 distribué sous licence BSD et basé sur le modèle MVC.



Installation

Seagull est distribué sous forme d'une archive à extraire dans le répertoire du projet.

Il utilise Pear pour son moteur de templates (Pear::Flexy), l'internationalisation (Pear::Translation2) et pour le mapping avec la base de données (Pear::DB_DataObjects). Pear doit donc être présent et les composants nécessaires installés.

Des modification de la configuration de PHP sont à prévoir car si Seagull fonctionne avec ou sans MagicQuotes, il est cependant recommandé d'allouer au moins 16 Mo pour l'exécution des scripts PHP.

Il est également nécessaire de modifier la configuration du serveur Web afin de limiter l'espace Web accessible au seul dossier www, ou à défaut de bloquer individuellement l'accès aux autres répertoires.



les frameworks PHP

Organisation des répertoires

▶ docs	/etc : divers scripts et schémas de base de données.
▶ etc	
▶ lib	/lib contient Seagull, Pear, ainsi que d'autres bibliothèques.
▶ modules	
▶ tests	
▶ var	/modules contient les différents modules activables dans seagull.
▼ www	
▶ images	
▶ js	/tests : dossier contenant les tests unitaires pour seagull et l'application.
▶ rpc	
▶ savant	
▶ smarty	/var : dossiers temporaires, cache, fichiers reçus.
▶ themes	
▶ tinyfck	/www : espace accessible par l'utilisateur. (images, javascript, ...)

Interface graphique

L'originalité de Seagull est de disposer dès l'installation d'une interface d'administration complète permettant de sélectionner les modules désirés (de nombreux modules sont disponibles dont CMS et Blog), gérer les comptes utilisateurs et les autorisation, créer les différentes pages du site et même disposer les différents blocs de contenu.

Seagull se présente donc comme une application très extensible disposant de fonctionnalités de base et d'une grande liberté de développement de modules.



les frameworks PHP

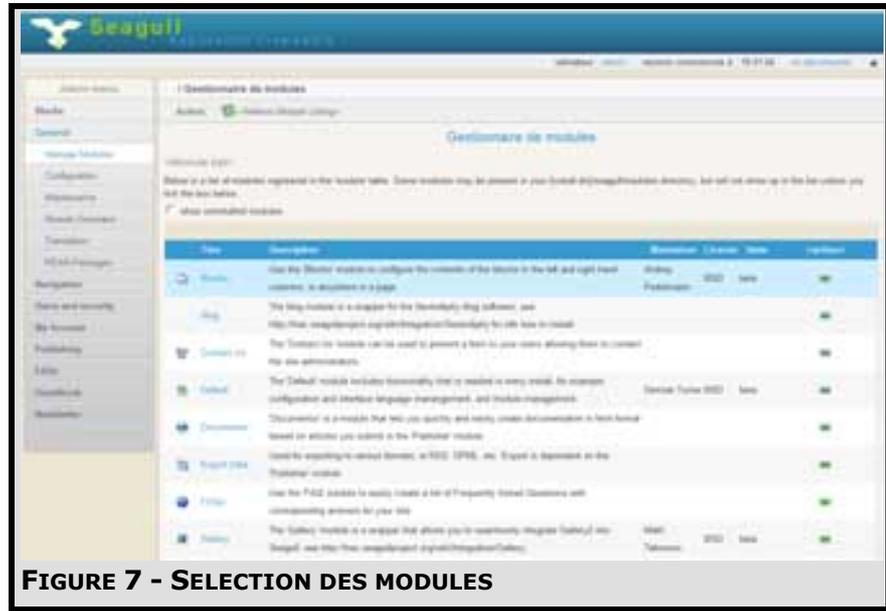


FIGURE 7 - SELECTION DES MODULES

www.smile.fr



FIGURE 1 - SEAGULL CMS

Templates

Seagull supporte 3 moteurs de templates : PEAR::Flexy , Smarty et Savant2.

Template Flexy :

```
<h1 class="pageTitle">Commentaires</h1>

<table >
  <tr>
    <th>Id</th><th>Titre</th>
  </tr>
  <tr flexy:foreach="comments[data],comment">
    <td>{comment[id]}</td><td>{comment[title]}</td>
  </tr>
</table>
```



les frameworks PHP

www.Mcours.com
 Site N°1 des Cours et Exercices Email: contact@mcours.com

```
</tr>
</table>
```

Génération de code

Un générateur de modules est fourni afin de créer facilement de nouvelles fonctionnalités.

Bases de données

Seagull utilise Pear pour la gestion de la base de donnée. Il supporte l'utilisation de plusieurs bases de données dans une même application. Il n'est par contre pas possible de préfixer le nom des tables afin de faire cohabiter proprement plusieurs applications.

Contrôleur :

```
<?php
require_once 'DB/DataObject.php';

class CommentsMgr extends SGL_Manager
{
    function CommentsMgr()
    {
        parent::SGL_Manager();
        $this->_aActionsMapping = array( 'list' => array('list'));
    }

    function validate($req, &$input)
    {
        //Action par défaut
        $input->action = ($req->get('action'))?$req->get('action'):'list';
        $this->validated = true;
    }

    function _cmd_list(&$input, &$output)
    {
        SGL::logMessage(null, PEAR_LOG_DEBUG);
        $output->template = 'commentsList.html';
        $output->pageTitle = 'CommentsMgr :: List';

        $query = "SELECT id, title FROM
            {$this->conf['table']['comments']}";

        $aData = SGL_DB::getPagedData($this->dbh,$query,
            $pager_options = array(),$disabled = true);

        if (PEAR::isError($aData)) {
            return false;
        }
        $output->comments = $aData;
    }
}
?>
```

Sécurité

Seagull intègre un système d'authentification de l'utilisateur et un système de gestion de droits. L'interface d'administration de Seagull inclut les pages de configuration correspondantes.



FIGURE 8 - GESTION DES PERMISSIONS

AJAX

Seagull ne facilite pas l'utilisation de requêtes AJAX.

Maturité

Malgré un numéro de version (0.6) faisant penser à un nouveau framework, Seagull existe depuis 2001. Son cycle de développement est très rapide, ce qui permet d'obtenir en permanence de nombreux retours de la part des utilisateurs.

La communauté autour du projet n'est pas très grande, mais la société à la source du projet propose un support commercial.

Conclusion

Seagull est un framework mature, qui est conseillé pour les applications qui incorporent un CMS simple, un blog, ou tout autre fonctionnalité déjà implémentée dans Seagull.

Il n'est en revanche pas idéal pour les application très spécifiques ou utilisant fortement l'Ajax.



les frameworks PHP

PHPonTrax

PHPonTrax est un portage de Ruby on Rails pour PHP 5. Le résultat est très fidèle, à tel point que le livre « Agile Web Development with Rails » s'applique directement à PHPonTrax et seuls les exemples ont besoin d'être modifiés.

PHPonTrax est distribué sous licence MIT (open source).



Installation

L'installation de PHPonTrax se fait via Pear. Des packages Pear sont utilisés (notamment Pear ::DB) et doivent donc être installés.

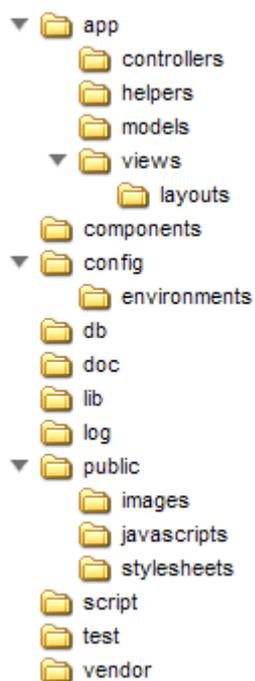
Il est ensuite nécessaire de créer un fichier bat ou sh à la main afin d'exécuter les commandes de génération de code.

Sous Windows la modification de plusieurs fichiers sera nécessaire pour faire fonctionner PHPonTrax.



les frameworks PHP

Organisation des répertoires



/app : contient l'application, on y retrouve la séparation modeles, vues, et controleurs.

/config : la configuration de l'application.

/doc : documentation de l'API de l'application, générée par PHPDoc.

/public : l'espace web accessible par l'utilisateur

/script : Les scripts de génération de code.

Templates

PHPonTrax utilise des templates au format PHP :

Template

```
<h1>Commentaires</h1>

<table border="0">
  <tr>
    <?php foreach($content_columns as $column): ?>
      <th><?php echo $column['human_name'] ?></th>
    <?php endforeach; ?>
  </tr>

  <?php foreach($Comments as $Comment): ?>
    <tr>
      <?php foreach($Comment->content_columns as $column): ?>
        <td>
          <?php echo $Comment->send($column['name']) ?>
        </td>
      <?php endforeach; ?>
    </tr>
  <?php endforeach; ?>
</table>
```



Commentaires

Id	Title	Created	Modified
1	Entrée 1 - Demo	2006-08-23 16:40:02	2006-08-23 16:40:02
2	Entrée 2 - Smile	2006-08-23 16:41:34	2006-08-23 16:41:34
3	Entrée 3 - Symfony	2006-08-23 16:44:04	2006-08-23 16:44:04

FIGURE 9 - EXEMPLE PHPONTRAX

Accès aux données

Comme Rails, PHPonTrax utilise une implémentation de ActiveRecord :

Modèle :

```
<?php
class Comment extends ActiveRecord {
}
?>
```

Controleur

```
<?php
class CommentController extends ApplicationController {

    function index() {
        $Comment = new Comment();
        $this->Comments = $Comment->find all();
        $this->content_columns = $Comment->content_columns;
    }

}
?>
```

Il est possible de faire du scaffolding sur les modèles ActiveRecord.

Autres

PHPonTrax n'inclut pas de cache, de fonctions d'internationalisation ou de sécurité.

Ces fonctions peuvent être ajoutées grâce à Pear, ezComponents ou ZendFramework.



les frameworks PHP

AJAX

PHPonTrax n'intègre pas de bibliothèques ou de fonction spécialisées pour le développement de sites AJAX, mais peut être facilement configuré pour traiter ce type de requêtes.

Documentation

La documentation des API est assez complète, mais la documentation globale du framework est assez légère. Par contre, la documentation de Ruby on Rails semble pouvoir s'appliquer à PHPonTrax avec assez peu de modifications.

On pourra également chercher de l'aide sur le forum, assez actif.

Conclusion

PHPonTrax est très proche de Ruby on Rails, il propose les fonctionnalités de bases que l'on peut attendre d'un framework de développement rapide d'applications, mais il lui manque des fonctionnalités complémentaires, comme l'internationalisation ou la sécurité.

L'activité autour du projet semble assez réduite. Cependant ce framework peut être utilisé dans des projets de petite envergure.



les frameworks PHP

CodeIgniter

Code Igniter est un framework MVC pour PHP4 et 5, inspiré de Ruby on Rails et distribué sous license opensource. Il insiste sur son coté léger et très performant.



Installation

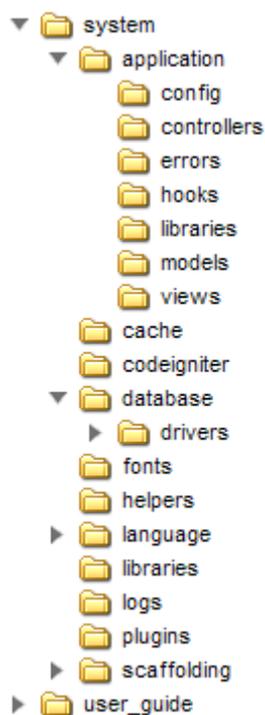
Code Igniter se présente sous la forme d'une archive à extraire dans le répertoire du projet.

La mise à jour n'est par contre pas aussi simple que dans les autres frameworks. La configuration de l'application n'étant pas dérivée d'une configuration globale, il est nécessaire d'appliquer manuellement certaines modifications lors de l'installation d'une nouvelle version.



les frameworks PHP

Organisation des répertoires



/system/application : l'application proprement dite : les contrôleurs, modèles, vues, bibliothèques utilisées et configuration de l'application se retrouvent dans leur dossiers respectifs.

/system/codeigniter : le framework en lui même

/system/libraries : les bibliothèques utilisées par CodeIgniter.

/system/plugins : plugins installés

/system/scaffolding : templates et images utilisées dans la génération des interfaces d'administration

/user_guide : documentation, disponible en local.

Template

Code Igniter utilise des templates au format PHP :

Template :

```

<table>
  <tr>
    <th>Id</th>
    <th>Titre</th>
    <th>Créé</th>
    <th>Modifié</th>
  </tr>
  <?php foreach( $query->result() as $row ) : ?>
    <tr>
      <td><?php echo $row->id?></td>
      <td><?php echo $row->title?></td>
      <td><?php echo $row->created; ?></td>
      <td><?php echo $row->modified; ?></td>
    </tr>
  <?php endforeach; ?>
</table>
  
```



Commentaires

Id	Titre	Créé	Modifié
1	Entrée 1 - Demo	2006-08-23 16:40:02	2006-08-23 16:40:02
2	Entrée 2 - Smile	2006-08-23 16:41:34	2006-08-23 16:41:34
3	Entrée 3 - Code Igniter	2006-08-23 16:44:04	2006-08-23 16:44:04

FIGURE 10 - EXEPLÉ CODE IGNITER

Accès aux données

Code Igniter utilise une implémentation d'ActiveRecord.

Modèle :

```
<?php
class Items_model extends Model {

    function Items_model() {
        parent::Model();
    }
}
?>
```

Contrôleur :

```
<?php
class Comments extends Controller {

    function Comments() {
        parent::Controller();
    }

    function index() {
        $data['query'] = $this->db->get( 'comments' ) ;
        $this->load->view( 'comments_view', $data ) ;
    }
}
?>
```

Le scaffolding est activé par l'ajout d'un mot clé (configurable) dans l'url lors de l'accès à un modèle sans contrôleur.

Un outil de débogage et de statistiques est intégré et permet d'obtenir les requêtes et le temps passé pour générer chaque page.

Cache

Code Igniter dispose d'un système de cache assez basique. Une fois celui-ci activés, les pages sont mises en cache pendant une certaine période.



les frameworks PHP

Documentation

La documentation de Code Igniter est très complète, mais comporte parfois des erreurs. Ainsi certaines parties du guide de démarrage rapide sont significativement différentes de celles contenues dans la documentation détaillée et ne permettront pas d'obtenir une application fonctionnelle.

Maturité

Code Igniter est un framework plutôt jeune et dynamique, dont la première version date de février 2006. Assez complet, même si il manque des fonctions telles que la gestion des droits ou l'internationalisation.

Conseillé pour les développements de petite envergure.



Copix

Copix est un framework PHP4 français, initié par Aston et développé depuis 2003.

Compatible depuis peu avec PHP5, il est basé sur un modèle MVC auquel s'ajoute un système d'événements.

Ce framework n'utilise pratiquement pas de conventions : le développeur doit par exemple déclarer les action et les modèles utilisés pour l'accès aux données.



Installation

Copix est livré sous forme d'archive à décompresser dans le répertoire projet.

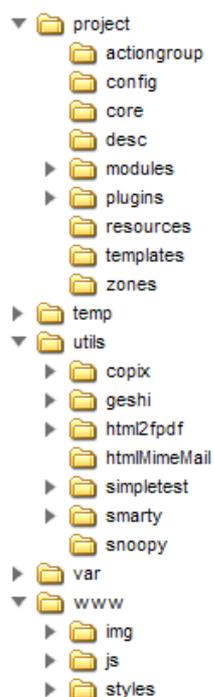
La désactivation des Magic Quotes est conseillée, mais optionnelle. Un plugin est disponible en cas d'impossibilité de modification de la configuration du serveur.

Il est possible de configurer Copix pour utiliser des URL significatives, mais cette fonctionnalité est pas considérée comme stable dans la version actuelle.



les frameworks PHP

Organisation des répertoires



/project : contient le code de l'application développées, les templates, la configuration et la liste des plugins activés.

/tmp : les fichiers temporaires et le cache.

/utils : le framework Copix ainsi que les bibliothèques nécessaires au projet.

/www : contient l'espace web public (images, css et javascript)

Copix peut être paramétré pour partager le dossier utils entre plusieurs applications.

Templates

Copix permet d'utiliser des templates PHP et Smarty.

Template Smarty:

```
<h1>Commentaires</h1>

<table>
  <tr>
    <th>Id</th>
    <th>Titre</th>
    <th>Créé</th>
    <th>Modifié</th>
  </tr>
  {foreach from=$comments item=comment}
    <tr>
      <td>{$comment->id}</td>
      <td>{$comment->title}</td>
      <td>{$comment->created}</td>
      <td>{$comment->modified}</td>
    </tr>
  {/foreach}
</table>
```



les frameworks PHP

Commentaires

Id	Titre	Créé	Modifié
1	Entrée 1 - Demo	2006-08-23 16:40:02	2006-08-23 16:40:02
2	Entrée 2 - Smile	2006-08-23 16:41:34	2006-08-23 16:41:34
3	Entrée 3 - Copix	2006-08-23 16:44:04	2006-08-23 16:44:04

FIGURE 11 - EXEMPLE COPIX

Des exemples de templates PHP se trouvent dans les frameworks étudiés précédemment.

Accès aux données

Copix utilise un ORM basé sur des DAO. La structure de chaque table doit être décrite dans un fichier XML. Il n'y a donc pas de détection automatique similaire à ActiveRecord ou Propel.

Il est également possible d'exécuter directement des requêtes au format SQL. Des classes sont disponibles afin de simplifier le développement de ces requêtes.

Plusieurs profils de base de données peuvent être définis ce qui permet d'accéder à des bases de données différentes depuis la même application.

Modèle :

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<daodefinition version="1">
  <datasource>
    <tables>
      <table name="comments" tablename="comments_demo"
              primary="yes" />
    </tables>
  </datasource>
  <properties>
    <property name="id" fieldName="id" type="autoincrement"
              pk="true" required="yes" />
    <property name="title" fieldName="title" type="string"
              required="yes" />
    <property name="created" fieldName="created"
              type="date" required="yes" />
    <property name="modified" fieldName="modified"
              type="date" required="yes" />
  </properties>
</daodefinition>
```



les frameworks PHP

```
</properties>
</daodéfinition>
```

Contrôleur :

```
<?php
/**
 * Page d'accueil et fonctionnalité standard.
 */
class ActionGroupComments extends CopixActionGroup {

    function getIndex () {
        //création de l'objet template.
        $tpl = new CopixTpl ();

        $DAOcomments = & CopixDAOFactory::create ("Comments");
        $liste_commentaires = $DAOcomments->findAll ();
        $tpl->assign ('comments', $liste_commentaires);

        //retour de la fonction.
        return new CopixActionReturn (COPIX_AR_DISPLAY_IN,
            $tpl, 'comments.tpl');
    }
}
?>
```

Définition de l'action :

Chaque action doit être déclarée dans Copix.

```
$comments = & new CopixAction ('comments', 'getIndex');
```

Cache

Un système de cache est disponible. Il est possible de définir plusieurs types de caches et de gérer indépendamment l'activation de chacun d'eux.

Ajax

Copix inclut les principales bibliothèques AJAX et comporte des fonctions pour faciliter leur utilisation.

Internationalisation

L'internationalisation est supportée via des fichiers de propriétés contenant les traductions des chaînes de caractères, ou des templates différentes pour chaque langue.



les frameworks PHP

Sécurité

Des plugins d'authentification et de gestion de droits sont intégrés à Copix.

Le système se base sur la définition de groupes et l'attribution de droits d'accès aux domaines et aux actions.

La liste des différents droits disponibles pour chaque action est fixe mais assez complète : none, show, read, write, valid, publish, moderate, admin, ce qui couvre la majorité des besoins.

Evénements

Copix intègre un système de gestion d'événements, ce qui est original pour un framework MVC.

Les modules peuvent écouter certains événements et réagir en conséquence.

Extensions

Copix dispose d'un système de plugins, l'installation en comporte déjà un certain nombre. On notera que certaines fonctionnalités de base du framework sont implémentés sous forme de plugin.

Documentation

Le site de Copix est assez pauvre en information mais permet de télécharger une documentation PDF très complète du framework. Un wiki et un forum sont également disponibles.

Maturité

Copix est un framework assez ancien. La communauté est plutôt réduite et principalement française. Le cycle de développement est par conséquent assez lent.

Cependant le framework est tout à fait mature et est utilisé dans plusieurs grands sites français.



les frameworks PHP

Conclusion

Copix est un framework mature, moins tourné vers le développement rapide d'application que les autres frameworks PHP. Il n'utilise pas de conventions ni de génération de code et demande plus de configuration au développeur. Il n'est pas non plus optimisé pour PHP5.

La prochaine version annoncée apportera une optimisation pour PHP5 ainsi que de nombreuses nouveautés.

S'il est parfaitement utilisable dans sa version actuelle, on aura peut-être avantage à attendre la sortie de la version 3, imminente au moment de l'écriture de ce livre blanc.



les frameworks PHP

Jelix

Jelix⁹ est un framework basé sur le code de Copix (fork) dans lequel tout le fonctionnement a été repensé et retravaillé.

Le framework a également été optimisé pour PHP5.

Cette présentation souligne les différences notables de Jelix par rapport à Copix.

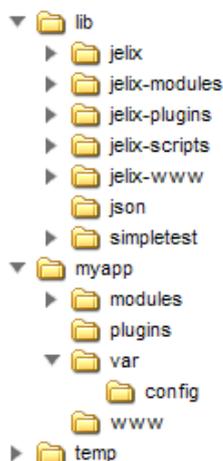


Installation

L'installation du framework se fait à partir d'une archive à décompresser. Un squelette d'application peut ensuite soit être généré ou installé via une seconde archive.

Comme Copix, Jelix demande la désactivation des Magic Quotes mais un plugin permet au framework de fonctionner si la désactivation n'est pas possible.

Organisation des répertoires



/lib : contient le framework Jelix.

/temp : utilisé pour les fichiers temporaire et le cache métier.

/myapp : l'application développée. Le nom du dossier est à remplacer par le nom de l'application.

⁹ <http://jelix.org>



les frameworks PHP

Génération de code

Jelix permet de générer une nouvelle application ou un nouveau module depuis une ligne de commande :

```
// Génération d'une application
jelix createapp

// Génération d'un module
jelix createmodule comments
```

Templates

Jelix utilise un moteur de template spécifique, jTpl :

Exemple de template jTpl :

```
<h1>Commentaires</h1>

<table>
  <tr>
    <th>Id</th>
    <th>Titre</th>
    <th>Créé</th>
    <th>Modifié</th>
  </tr>
  {foreach $comments as $comment}
    <tr>
      <td>{$comment->id}</td>
      <td>{$comment->title}</td>
      <td>{$comment->created}</td>
      <td>{$comment->modified}</td>
    </tr>
  {/foreach}
</table>
```

Commentaires

Id	Titre	Créé	Modifié
1	Entrée 1 - Demo	2006-08-23 16:40:02	2006-08-23 16:40:02
2	Entrée 2 - Smile	2006-08-23 16:41:34	2006-08-23 16:41:34
3	Entrée 3 - Jelix	2006-08-23 16:44:04	2006-08-23 16:44:04

FIGURE 12 - EXEMPLE JELIX

Accès aux données

Comme Copix, Jelix utilise des objets DAO pour accéder aux données. Ces objets nécessitent l'écriture d'un descriptif XML.



les frameworks PHP

Jelix apporte la possibilité de générer automatiquement ce fichier XML à partir des informations de la base de données.

Génération d'un DAO :

```
jelix createdao comments comments comments
```

Modèle :

```
<?xml version="1.0" encoding="iso-8859-1"?>
<dao xmlns="http://jelix.org/ns/dao/1.0">
  <datasources>
    <primarytable name="comments" realname="comments"
      primarykey="id" />
  </datasources>

  <record>
    <property name="id" fieldname="id"
      datatype="bigautoincrement" required="yes"/>

    <property name="title" fieldname="title"
      datatype="string" required="yes"/>

    <property name="created" fieldname="created"
      datatype="date"/>

    <property name="modified" fieldname="modified"
      datatype="date"/>

  </record>
</dao>
```

Contrôleur :

```
<?php
class CTdefault extends jController {

  function index() {
    $rep = $this->getResponse('html');
    $rep->title = 'Commentaires';
    $rep->bodyTpl = 'commentaires';

    $comments = jDao::get('comments~comments');
    $commentsList = $comments->findAll();
    $rep->body->assign('comments', $commentsList);

    return $rep;
  }
}
?>
```



les frameworks PHP

Documentation

La documentation présente sur le site officiel est plutôt complète, et contient également des tutoriaux ainsi qu'une documentation de l'API.

Maturité

Jelix est un framework très jeune. Même s'il est basé sur Copix, il a été suffisamment modifié pour –potentiellement– perdre en stabilité. La communauté autour du projet est également très réduite.

Conclusion

Jelix est en quelques sorte un Copix amélioré. Il n'a cependant qu'une seule référence à l'heure actuelle : <http://www.kayuko.com/>

Certainement meilleur que Copix 2.3 au niveau des fonctionnalités et de la vitesse, il reste trop jeune pour l'utiliser dans des projets importants.



les frameworks PHP

WACT

WACT est un framework MVC pour PHP5.

Il n'y a pas eu de version stable depuis la sortie de la version 0.2 2004. WACT est uniquement disponible en version de développement.



fin

Installation

Afin d'installer WACT, il est nécessaire d'obtenir la version de développement depuis le référentiel SVN du projet.

Documentation

La documentation se réfère à la version 0.2. La version de développement n'est pas documentée.

Conclusion

WACT est un framework à déconseiller. Même s'il est utilisé avec succès sur quelques sites, il n'a pas la maturité que l'on attend d'un framework : il n'y a pas eu de versions stable depuis plusieurs années, les fonctionnalités couvertes se réduisent au minimum et la communauté est très réduite.



les frameworks PHP

Zoop

Zoop est un framework PHP4 utilisant un modèle MVC et distribué sous licence ZPL (compatible avec la licence GPL).



Installation

Zoop est distribué sous forme de deux archives : le framework à installer en dehors de l'espace web et un squelette d'application. La mise à jour consiste à simplement remplacer le framework par la nouvelle version.

Zoop utilise des bibliothèques Pear qu'il faudra installer manuellement. Il prend également en charge les serveurs où Pear n'est pas accessible.

Une fois le squelette d'application installé, il faudra spécifier le chemin d'accès du framework dans la configuration.

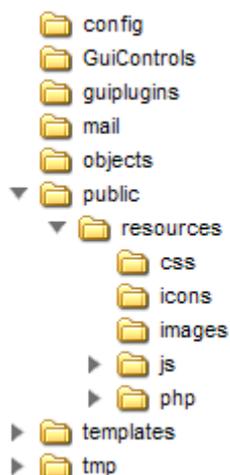
D'autres modifications sont également nécessaires pour faire fonctionner le framework sous Windows.

Dans sa version actuelle, Zoop utilise les shorts tags PHP (< ? au lieu de < ?php) qu'il faudra donc activer dans la configuration du serveur.



les frameworks PHP

Organisation des répertoires



Le framework Zoop s'installe en dehors de l'arborescence du projet.

/config : la configuration de l'application.

/public : l'espace web accessible aux utilisateurs.

/templates : les templates utilisées par l'application.

Templates

Zoop utilise le moteur de templates Smarty. L'appel d'une template ne nécessite que 2 lignes de PHP.

Exemple de template :

```

<h1>Commentaires</h1>

<table>
  <tr>
    <th>Id</th>
    <th>Titre</th>
    <th>Créé</th>
    <th>Modifié</th>
  </tr>
  {foreach from=$comments item=comment}
    <tr>
      <td>{$comment.id}</td>
      <td>{$comment.title}</td>
      <td>{$comment.created}</td>
      <td>{$comment.modified}</td>
    </tr>
  {/foreach}
</table>
  
```



les frameworks PHP

Commentaires

Id	Titre	Créé	Modifié
1	Entrée 1 - Demo	2006-08-23 16:40:02	2006-08-23 16:40:02
2	Entrée 2 - Smile	2006-08-23 16:41:34	2006-08-23 16:41:34
3	Entrée 3 - Zoop	2006-08-23 16:44:04	2006-08-23 16:44:04

FIGURE 13 – EXEMPLE ZOOP

Zoop propose également des composants « GuiControls » qui permettent de faciliter la création de formulaires ainsi que la vérification des données entrées par les utilisateurs :

Exemple GuiControls

```
<form method="POST">
  <p>
    <label for="controls[text][from][value]">From:</label>
    {guicontrol type='text'
      name='from'
      _validate_type='email'
      _validate_required='true'}<br>

    <label for="controls[text][subject][value]">Subject:</label>
    {guicontrol type='text'
      name='subject'
      _validate_type='length'
      _validate_min=5
      _validate_max=25
      _validate_required='true'}<br>

    <label for="controls[text][message][value]">Message:</label>
    <br> {guicontrol type='textarea'
      name='message'
      _validate_required='true'}<br>

    <br>
    <input type="submit" onclick="return submitForm(form);">
  </form>
```

Cependant cette fonctionnalité fait perdre l'avantage principal des templates qui est de séparer la partie fonctionnelle et la partie présentation de l'application. Ici, la modification d'une template influence directement le format des données soumises à l'application.

Certains contrôles, notamment Slider, posent des problèmes lorsque le navigateur utilisé n'est pas Internet Explorer.



les frameworks PHP

Forms2

La fonctionnalité Forms2 permet de faciliter l'affichage et l'édition de données contenues en base de données.

Exemple :

```
function pageListUsers($inPath)
{
    global $gui;
    $form = new form2("users", "list");
    $form->setParam("zone", $this->zonename);
    $form->setTitle("Viewing Users");
    $form->setParam("listlink", "EditUser");
    $form->setFieldParam(array("password", "info"), "listshow",
false);
    $form->setFieldParam(
array("first_name", "username", "last_name"),
"clickable", true);
    $form->setFieldIndex("activated", array(1 => "y", 0 => ""));
    $form->setParam("deleteColumn", true);
    $form->setParam("deletelink", "DeleteUser");

    $form->guiAssign();

    $gui->generate("forms/form2.tpl", "blank.tpl", "mainmenu.tpl",
"Users");
}
```

Ce système ressemble donc à une sorte de scaffolding, mais c'est à l'utilisateur de développer les différents écrans.

Userid	First Name	Last Name	Email Address	Username	Last Login	Activated	Del
1	Steve	Tester	test@testing.com	Tester	0000-00-00 00:00:00		X
2	John	Tester	testingrocks@testing.com	test4me	0000-00-00 00:00:00	y	X

FIGURE 14 - FORMS2

Accès aux données

Zoop ne dispose pas de couche ORM. Il propose cependant une couche d'indépendance de la base de données basée sur Pear :DB.

Controleur MVC

Zoop utilise une notion de « zones » qui correspond aux « routes » que l'on retrouve dans un framework MVC classique.



les frameworks PHP

Fonctionnalité supplémentaire, les zones permettent de définir des fonctions qui seront exécutées pour toutes les actions appartenant à cette zone.

Contrôleur :

```
<?
class zone_default extends zone
{
    function pageDefault($inPath)
    {
        global $gui;

        $result = sql_fetch_rows("select * from comments");
        $gui->assign("comments", $result);

        $gui->display("comments.tpl");
    }
}
?>
```

Cache

Zoop dispose d'un système de cache qu'il est possible d'utiliser pour stocker des variables ou le résultat de requêtes SQL.

Maturité

Zoop est un framework mature qui existe depuis 2003. Il souffre cependant d'une communauté assez réduite.

Documentation

La documentation est complète et comprend plusieurs tutoriaux.

Conclusion

Bien qu'il soit mature et utilisé par plusieurs sites, Zoop souffre du manque de certaines fonctionnalités telles que l'ORM, l'internationalisation, le scaffolding, les conventions. L'implémentation de la validation des formulaires dans les templates est également discutable.

On recommandera l'utilisation d'un autre framework.



les frameworks PHP

TABLEAU COMPARATIF

	PHP	Modèle	ORM	Templates	Cache	URL conviviales	Validation Formulaires	Ajax	Extensions	Génération de code	Internationalisation I18n
PEAR	4,5	/	DB_DataObject	Flexy, ...	O	N	O	N	N	N	O
ezComponents	5.	/	DAO	ezTemplate	O	N	N	N	N	N	O
Zend Framework	5	MVC	/	PHP	O	O	N	N	N	N	N
CakePHP	4,5	MVC	AR	PHP	O	O	O	O	O	O	N
Symfony	5	MVC	Propel	PHP ou Smarty	O	O	O	O	O	O	O
Prado	5	Event	/	Prado	O	O	O	O	O	N	O
Seagull	4,5	MVC	Pear	Flexy, Smart ou Savant2	O	N	O	N	O	O	O
PhpOnTrax	5	MVC	AR	PHP	N	N	O	O	N	O	N
CodeIgniter	4,5	MVC	AR	PHP	O	O	O	N	O	N	N
Copix	4	MVC	DAO	Smarty ou PHP	O	O	N	N	O	N	O
Jelix	5	MVC	DAO	jTpl	O	O	O	O	O	O	O
Wact	4,5	MVC	/	wact	O	N	O	N	N	N	N
Zoop	4,5	MVC	/	Smarty	O	O	O	O	O	N	N

O : Oui , **N** : Non, **AR** : Active Record



les frameworks PHP



CONCLUSION

L'utilisation d'un framework est souvent indispensable pour les développements de moyenne ou grande envergure, mais représente également un intérêt pour les petits développements.

La phase d'apprentissage du framework sera courte dans la plupart des cas et le développeur pourra vite tirer parti de ses avantages. Le retour sur investissement est rapide.

Quels frameworks choisir pour vos projets PHP ?

Comme nous l'avons dit en introduction les véritables frameworks sont fortement structurants, ils définissent l'architecture logicielle de l'application. Il n'est donc pas possible de les mettre en œuvre sur des projets anciens, sauf à tout redévelopper.

C'est pourquoi notre recommandation sera différente selon que l'on considère un nouveau projet, ou bien l'enrichissement d'une application existante.

Pour un nouveau projet, les frameworks qui se distinguent sont **CakePHP**, **Symfony**, et **Prado**. Ils offrent la meilleure combinaison de maturité et de fonctionnalité, en même temps qu'une garantie satisfaisante de pérennité et de support communautaire. Utilisés avec PHP5, ils permettent d'appréhender de grands projets et démontrent que le langage PHP n'est plus réservé aux petits développements.

Pour l'enrichissement ou la maintenance d'applications existantes, on retiendra plutôt les bibliothèques de composants de **eZ Components** ou **ZendFramework** qui permettront de rajouter rapidement des fonctionnalités robustes et parfaitement implémentées.

Parmi les autres produits, certains ont une couverture fonctionnelle insuffisante, d'autres sont prometteurs, mais pas encore mûrs pour une utilisation opérationnelle, d'autres enfin sont au contraire un peu vieillissants et n'intègrent pas les approches les plus modernes du développement.