



CONSULTING

avec



The PHP Company

Zend Framework

Trouver le chemin des bonnes pratiques

Webinar Zend Technologies
25 Octobre 2011

[voir le webinar complet en ligne](#)

Gauthier Delamarre, Professional Services Manager pour VA Consulting - www.vaconsulting.lu

Vous avez dit Zend Framework ?

Donc, vous avez dit framework...

- ▶ Ce qu'est un framework
 - un ensemble de composants
 - ... et de bonnes pratiques
 - ... ainsi que de standards (nommage et codage)
- ▶ Ce que n'est pas un framework
 - ~~un atelier de développement rapide (RAD)~~
 - ~~un générateur de code~~

et Zend Framework ?

▶ Objet

- le code de Zend Framework est 100% objet

▶ Technologie

- Zend Framework exploite le meilleur de PHP

▶ Souplesse

- les liens entre les composants sont faibles

▶ Communauté

- très importante, encadrée, milieu industriel

Choisir Zend Framework

- ▶ **Raisons internes**
 - historique
 - compétences pré-existantes
 - besoins du projet
- ▶ **Raisons externes**
 - ressources disponibles
 - formation
 - accompagnement

Avant de démarrer un projet...

Adopter le bon état d'esprit

- ▶ **Un framework est un outil, pas un concurrent**
 - ne pas s'interroger sur la pertinence des composants une fois le framework choisi
- ▶ **Se concentrer sur le métier**
 - la valeur ajoutée d'un projet est portée par la logique métier
 - l'architecture, qui n'a aucune valeur ajoutée, est confiée au framework

Se mettre à niveau

- ▶ S'assurer de maîtriser les pré-requis techniques
 - particulièrement l'OOP
 - interfaces
 - classes abstraites
- ▶ Acquérir une bonne maîtrise du framework
 - connaître les composants existants
 - connaître les standards applicables
- ▶ Dans les deux cas, il faut souvent se former

Premiers pas avec ZF

Choisir l'arborescence du projet

▶ Mode simple

- MVC unique
- peut être généré par Zend_Tool
- réservé aux projets modestes

▶ Architecture modulaire

- MVC multiple
 - une arborescence MVC par module
- adaptée aux projets plus vastes
- facilite la réutilisabilité

En parlant de réutilisabilité

- ▶ ZF offre différents mécanismes de plugins
 - helpers de vues et d'actions
 - plugin de front controller
 - form elements, decorators...
- ▶ Distinguer les plugins génériques des plugins métiers
 - créer un dépôt global pour les génériques
 - ... et un autre dans l'arborescence du projet pour les plugins liés au métier

Définition des standards

- ▶ Les standards de nommage et de codage garantissent la lisibilité du code
 - leur respect facilite la prise en main du projet par un tiers : consultant, nouveau collaborateur
- ▶ ZF applique lui-même certains de ces standards
 - il est plus simple de s'y conformer (et parfois obligatoire)
 - les utiliser facilite également l'intervention d'un tiers

Bonnes pratiques «illustrées»

Controllers

▶ Bonne pratique

- les contrôleurs ne doivent contenir que des méthodes d'actions

▶ Raisons

- créer une méthode dans un contrôleur limite son utilisation à ce seul contrôleur
- interdit toute réutilisation de cette méthode dans un autre projet

▶ Implémentation

- créer des actions helpers

Models

▶ Bonne pratique

- ne pas mélanger la logique métier et l'accès aux données

▶ Raisons

- les objets métiers qui accèdent aux données ne peuvent pas être testés unitairement
- changer de couche d'accès aux données peut s'avérer très compliqué

▶ Implémentation

- scinder les modèles entre métier et données

Views

▶ Bonne pratique

- ne pas créer de dépendances entre vues et objets métiers

▶ Raisons

- une telle dépendance empêche de paralléliser les tâches
- une modification de l'objet métier implique un contrôle de toute les vues qui l'utilisent

▶ Implémentation

- passer des tableaux de données aux vues

Formulaires

▶ Bonnes pratique

- utiliser à 100% le mécanisme de rendu HTML des formulaires (via les décorateurs)

▶ Raisons

- l'affichage d'un formulaire est dépendant de son état - les décorateurs en tiennent compte automatiquement
- conserve la souplesse du système de plugins

▶ Implémentation

- `echo $this->form;` dans les vues !

Conclusion / Questions

Merci de votre participation !

[voir le webinar complet en ligne](#)