

# Zend Framework - Organisation des dossiers

par JYT ([Les expériences Zend de Sekaijin](#)) (Blog)

Date de publication : 10 octobre 2007

Dernière mise à jour :

Une des premières choses à faire lorsqu'on commence à coder une application est de définir la hiérarchie des dossiers.  
Y réfléchir avant de commencer permet d'éviter la pagaille qui attend les fichiers au fil du temps si on n'y prend garde.  
Une chose est certaine, il n'existe pas d'organisation qui réponde à tous les besoins. Mais on peut être tout aussi assuré, que si on ne définit rien, au fil du temps, il va être difficile de retrouver ses petits.

I - Structure d'une Application web.....	3
II - Et la sécurité ?.....	4
III - Une première approche.....	5
IV - Et les modules ?.....	6
V - Une proposition d'organisation.....	7

## I - Structure d'une Application web

Mon expérience m'a conduit à définir trois grandes parties dans une application web.

Une première séparation apparaît entre ce qui est purement du ressort du web et ce qui est applicatif. On trouve immanquablement dans une web application des éléments directement accessibles par le client : les images, feuilles de styles, *etc.* Tous ces éléments ont juste à se trouver dans un dossier visible du serveur HTTP.

De l'autre côté de la frontière, on trouve tout ce qui a besoin du moteur applicatif pour être exploitable.

Parmi ces derniers éléments, on peut encore trouver une séparation entre tous les éléments réutilisables et ceux qui sont propres à l'application que l'on développe.

Zend Framework propose une organisation tout en laissant libre le développeur de définir la sienne.

## II - Et la sécurité ?

Une question importante est de définir ce que nous devons ou ne devons pas mettre dans l'arborescence HTTP. *A priori*, une bibliothèque de fonctions ne devrait pas se trouver dans cette hiérarchie. Seuls les éléments visibles du client le devraient. Malheureusement, il m'est arrivé à plusieurs reprises de ne pouvoir écarter des fichiers de l'espace public, l'hébergement ne l'autorisant pas. Il convient donc de protéger ce qui doit l'être de toute tentative.

Au final, j'ai fini par tout avoir dans l'espace web. Ainsi, quel que soit l'hébergement, je suis en mesure de déployer mon application. Le choix d'une organisation *a priori* impose de s'être une bonne fois pour toutes posé cette question de la sécurité des ressources.

### III - Une première approche

J'ai pour ma part organisé mes développements ainsi :

```
/
  application/
  library/
  public/
```

Travaillant en équipe, la normalisation de l'organisation est une source de gain. Ainsi, lorsqu'on reprend l'application d'un autre, on y voit clair. Mais cette première hiérarchie est un peu trop basique.

Partant des ajustements que nous avons faits au fil du temps, j'ai organisé le dossier **public** en sous-dossiers en fonction du type de contenu : images, styles, scripts, medias.

Pour le dossier **application**, j'ai suivi les recommandations de ZF. Ce n'était pas bien compliqué vu qu'ils correspondaient aux noms près à l'organisation que j'avais déjà mise en place dans le passé.

Quant au dossier **library**, il contient la bibliothèque Zend bien sûr et d'autres bibliothèques organisées en fonction de leur origine.

Pour tout ce qui est des éléments que j'ai été amené à ajouter, je les ai regroupés dans la bibliothèque **Fast**. Non pas que ma bibliothèque soit rapide mais elle a pour but d'accélérer les développements.

## IV - Et les modules ?

Restait un petit problème d'organisation. ZF nous propose de mettre nos modules applicatifs dans le dossier **application**. C'est une bonne chose que je pratique volontiers. Mais un des avantages de faire des modules c'est de pouvoir les réutiliser d'une application sur l'autre. Il existe des modules qui n'ont d'autre intérêt que pour l'application qui lui a donné naissance, mais il est possible d'en avoir qui soient suffisamment génériques pour être réutilisables. Je me suis donc demandé s'il était opportun de les conserver dans le dossier **application**.

Après mûres réflexions, on peut voir ces modules comme de mini applications dans l'application. J'ai donc décidé de les sortir du dossier **application**. Là encore, pour bien identifier ceux que je développe, j'ai décidé de les regrouper dans un dossier.

## V - Une proposition d'organisation

Au final je suis arrivé à l'organisation suivante :

```
/
  application/
    Controllers/
    Models/
    Views/
  fast_modules/
    module1/
      Controllers/
      Models/
      Views/
  index.php
  library/
    Fast/
    Zend/
  public/
    Images/
    Scripts/
    Styles/
```

Chaque dossier ayant un rôle bien identifié, l'insérer dans l'organisation devient relativement aisé. Mais il ne faut pas perdre de vue qu'on ne pourra éviter qu'un jour, on tombera inmanquablement sur un élément qui nous posera des problèmes.

Dans toute cette organisation, seuls **index.php** et le dossier **public** doivent être visibles du client. Tout le reste doit être protégé. Une bonne configuration du serveur fait le nécessaire.

Ainsi paré, je n'ai plus à me préoccuper de l'organisation de mes applications. Elles sont toutes bâties sur ce modèle.

Pour ceux qui commencent avec ZF, ne vous inquiétez pas trop de la quantité de dossiers ici présents. Je les passerai en revue au fils de mes posts. Pour la plupart, vous les trouverez dans la documentation de ZF.