

# Zend Framework 2

---

## Les Médicaments

**Zend Framework 2 – Les médicaments Bouaziz – Philippe Traing – Yohann Teisseire – José Albea**  
**07/02/2014**

## Table des matières

Table des matières .....	1
Introduction.....	2
I. Liste de fonctionnalités implémentées: .....	3
II. Répartition des tâches.....	4
III. Optimisations mises en place :.....	4
IV. Scénarios de tests.....	5
V. Avantages et inconvénients de Zend Framework 2 .....	5
A. Avantages : .....	5
B. Inconvénients : .....	6
VI. Conclusion .....	6

## Introduction

De nos jours, il est de plus en plus rare qu'un développeur ait à créer un site web ou une application web sans utiliser de Framework. En effet, il existe une multitude de Frameworks disponibles aujourd'hui, fournissant des outils facilitant grandement le développement. Zend Framework est un des Framework les plus utilisés, y compris dans les grandes entreprises telles que la BNP Paribas, ou encore le site de la chaîne télévisée BBC. C'est pourquoi il est indispensable aujourd'hui de maîtriser au moins un Framework.

L'objectif de ce projet est donc l'acquisition d'une expérience en développement avec le Framework Zend Framework 2 en mettant en œuvre une application web sur les médicaments basée sur ce-dernier pour répondre aux besoins d'un client fictif.

## I. Liste de fonctionnalités implémentées:

Fonctionnalité	Composants utilisés
Création d'un projet via la console Zend Framework	Zend Console ZFTool
Création de l'architecture du projet via Zend Framework	
Création d'un module « Médicaments »	
Utilisation d'un fichier de configuration	Zend\Config
Choix du type de configuration via une liste déroulante proposant plusieurs types de fichiers : <ul style="list-style-type: none"> <li>Configuration via un fichier PHP</li> <li>Configuration via un fichier INI</li> <li>Configuration via un fichier XML</li> <li>Configuration via un fichier YAML</li> <li>Configuration via un fichier JSON</li> </ul>	Zend\Config Zend\Session  Zend\Config Zend\Config\Reader\Ini Zend\Config\Reader\Xml Zend\Config\Reader\Yaml Zend\Config\Reader\Json
Affichage de la liste des médicaments	Zend\Db\TableGateway\TableGateway
Ajout d'un médicament	Zend\Form\Form Zend\Form\Element Zend\InputFilter\Factory Zend\InputFilter\InputFilter Zend\Db\TableGateway\TableGateway
Modification d'un médicament	Zend\Form\Form Zend\Form\Element Zend\InputFilter\Factory Zend\InputFilter\InputFilter Zend\Db\TableGateway\TableGateway
Suppression d'un médicament	Zend\Form\Form Zend\Form\Element Zend\Db\TableGateway\TableGateway
Affichage des contrôleurs, vues et models en cours au niveau du pied de page	Zend\Debug
Conservation des noms de contrôleurs et vues parcourus dans un fichier de logs	Zend\Log
Gestion du multilinguisme afin de naviguer entre 4 langues : <ul style="list-style-type: none"> <li>Français</li> <li>Espagnol</li> <li>Anglais</li> <li>Allemand</li> </ul>	Zend\I18n
Diffusion des médicaments via flux RSS	Zend\Feed
Création d'un formulaire de contact	Zend\Form
Validation du formulaire + ajout captcha	Zend\InputFilter Zend\Captcha

## II. Répartition des tâches

Afin d'optimiser le temps de développement du projet, les tâches ont été réparties entre les quatre membres du groupe de la façon suivante :

- Création d'un projet via la console Zend Framework : **Philippe Traing**
- Création de l'architecture du projet via Zend Framework : **Philippe Traing**
- Création d'un module « Médicaments » : **Philippe Traing**
- Utilisation d'un fichier de configuration : **José Albea**
- Choix du type de configuration via une liste déroulante proposant plusieurs types de fichiers : **José Albea**
- Affichage de la liste des médicaments : **Axel Bouaziz**
- Ajout d'un médicament : **Axel Bouaziz**
- Modification d'un médicament : **Axel Bouaziz**
- Suppression d'un médicament : **Axel Bouaziz**
- Affichage des contrôleurs, vues et models en cours au niveau du pied de page : **Axel Bouaziz**
- Conservation des noms de contrôleurs et vues parcourus dans un fichier de logs : **Axel Bouaziz**
- Gestion du multilinguisme afin de naviguer entre 4 langues : **Yohann Teisseire**
- Diffusion des médicaments via flux RSS : **Axel Bouaziz**
- Création d'un formulaire de contact : **Yohann Teisseire + Philippe Traing**
- Validation du formulaire + ajout captcha : **Yohann Teisseire**

## III. Optimisations mises en place :

Zend Framework 2 possède un outil très pratique nommé ZFTool. Grâce à ce composant, nous avons pu en quelques lignes de commande seulement créer toute l'architecture Zend. Cette démarche a été un gain de temps considérable sur l'installation du projet puisqu'en quelques secondes, le projet était téléchargé et installé via le composant ZFTool.

De plus, ce composant nous a été utile tout le long du développement du projet. En effet, une simple ligne de commande nous a permis de créer toute l'architecture de notre module « Médicaments ». Par la suite, ayant vu la rapidité avec laquelle ZFTool permet de créer tout type d'actions, et afin d'éviter toute erreur, nous avons décidé de poursuivre le développement du projet avec cet outil. Par exemple, en tapant simplement « `php zftool.phar create controller NomDuController NomDuModule` » dans la console, un controller se crée instantanément dans le bon répertoire, avec les imports nécessaires et le répertoire pour la vue se crée avec l'action index par défaut. Il en était de même pour toutes les actions créées.

Une des fonctionnalités à implémenter était celle de l'utilisation d'une barre de debug en bas de page affichant les noms des contrôleurs, vues et models utilisés. Afin d'optimiser le code, nous nous sommes servis de la fonction « `onBootstrap` » présente dans le fichier `Module.php` du module Médicaments afin de transmettre ces données à chaque vue automatiquement. En effet, sans passer par cette fonction, il aurait été plutôt compliqué d'implémenter cette fonctionnalité puisqu'il aurait

fallu, dans chaque action créée, de chaque contrôleur, ajouter l'envoi de ces données vers le layout, et ce en passant exactement les mêmes variables à chaque action. Cette méthode n'était pas du tout viable, car elle impliquait qu'une personne reprenant le projet ne connaissant pas cette fonctionnalité ne puisse pas transmettre ces données ou commettre des erreurs.

Enfin, en ce qui concerne le multilinguisme, n'ayant pas plusieurs url d'accès au projet, nous avons préféré l'utilisation de cookies. Ainsi, par défaut la langue utilisée sera le Français, et si l'utilisateur décide de changer de langue, un cookie sera créé avec le code de la langue à utiliser et aura une durée de vie de 120 jours.

## **IV. Scénarios de tests**

Afin de valider chaque étape du développement, nous avons mis en place une procédure de recette. En effet, n'ayant jamais utilisé PHPUnit, nous n'avons pas réussi à implémenter cet outil. De plus, nous estimions que le temps d'apprentissage de PHPUnit et son utilisation pour chaque fonctionnalité était bien trop lourd pour un petit projet. C'est pourquoi les tests ont été réalisés manuellement.

Les étapes de tests étaient donc les suivantes : nous développons la fonctionnalité, puis vérifions son bon fonctionnement. Une fois le bon fonctionnement validé venaient les tests de sécurité. En effet, il était important par exemple, lorsqu'un identifiant est passé en paramètre dans l'URL, que ce paramètre soit bien de type integer, qu'il existe bien et qu'il rentre bien dans les identifiants existants. Ainsi, une batterie de tests était effectuée en entrant des identifiants de type textes, ou des identifiants non-existants pour être sûr que ce type d'intrusion ne posait pas de problème sur le site.

Un autre test crucial était mis en place pour tous les formulaires : celui de l'injection SQL. Pour bloquer les scripts effectuant des injections SQL, nous avons pris la précaution d'échapper tous les champs via la validation de formulaires de Zend, en utilisant les fonctions StrigTags, StringTrim. De cette manière, tous les codes html, JavaScript ou autre insérés sont retirés lors de l'envoi des données à la base de données.

## **V. Avantages et inconvénients de Zend Framework 2**

### **A. Avantages :**

Zend Framework 2 possède un très grand nombre d'avantages. En effet, comme nous expliqué ci-dessus, il est de plus en plus rare de devoir développer un site dit « from scratch ». De plus, Zend Framework est un Framework éprouvé, possédant une documentation et une forte communauté. Ces avantages font aussi parti de la force de Zend Framework. Grâce à ce-dernier, un très grand nombre d'étapes du développement sont réduites à quelques lignes de codes, et permettent de sécuriser le code bien plus facilement que si nous devions le faire nous-même.

## B. Inconvénients :

Malgré les grands avantages de Zend, ce Framework a aussi beaucoup d'inconvénients. En effet, la prise en main peut être très difficile pour un néophyte. L'installation en elle-même requiert l'utilisation de lignes de commandes, ce qui n'est pas accessible à tous et peut nécessiter beaucoup de temps si le développeur en charge de l'installation ne connaît pas ce principe.

La documentation de Zend a beau être très complète, cette-dernière n'est pas disponible en français pour la version 2 du Framework. De plus, cette-dernière est souvent trop vague. En effet, même si certains morceaux de codes expliquant les fonctionnalités sont présents, ces-derniers ne s'avèrent pas toujours fonctionnels.

Enfin, la différence entre les versions 1 et 2 de Zend étant très différentes, le temps d'apprentissage requis pour développer un petit projet est beaucoup trop long par rapport au temps qu'aurait pris ce même développement sans Framework.

## VI. Conclusion

En conclusion, ce projet nous a été d'une grande utilité puisqu'il nous a appris l'installation et l'utilisation d'un Framework très utilisé dans le monde des entreprises. De plus, certains d'entre nous n'avaient jamais utilisé de Framework jusqu'alors, ce qui nous a permis d'avoir un atout supplémentaire pour la poursuite de nos études, et plus tard, pour le monde professionnel.