**Mignonat Laurent** N° auditeur : PAC185806
Auditeur CNAM Paca (Aix-En-Provence) N° INE CNAM : 0g5drm9ad88

NFE102

Année 2013-2014, Semestre 2

# Projet de VideoClub

# I) Avancement du projet :

Les fonctionnalités du périmètre de l'application sont toutes implémentées. Il y a en plus l'authentification, et la gestion des droits en fonction du type d'utilisateur. Je n'ai pas eu le temps de consommer des webServices (2 autres UEs en parallèle).

# II) Réalisation du projet :

J'ai voulu installer mon environnement sans utiliser de solutions clé en main pour me familiariser avec mon nouvel OS, que je ne connaissait pas au début du projet (Mon premier mac avec Mac OS X). Je me suis donc lancé dans le paramétrage du vhost sur le serveur apache2 installé par défaut sur Mac OS X (pas si facile en fin de compte). J'ai aussi du installer MySql et PhpMyAdmin (très simple). C'étais plutôt sympa, surtout que j'ai pu tout faire en ligne de commande comme avec une distribution Linux.

J'ai codé avec NetBeans (j'ai laissé le répertoire NetBeansMetadata au cas ou vous voudriez éditer mes sources avec cet IDE). Pour tester le bon fonctionnement des liens de l'application j'ai déployé sur mon site : http://videoclub.mignonat.fr. (Login : **admin@video.club** / password : **azerty**)

J'ai déjà développé en Symfony2 au travail, j'ai donc plutôt facilement appréhendé le framework Zend. J'ai quand même du faire un petit effort pour m'adapter aux spécificités de Zend.

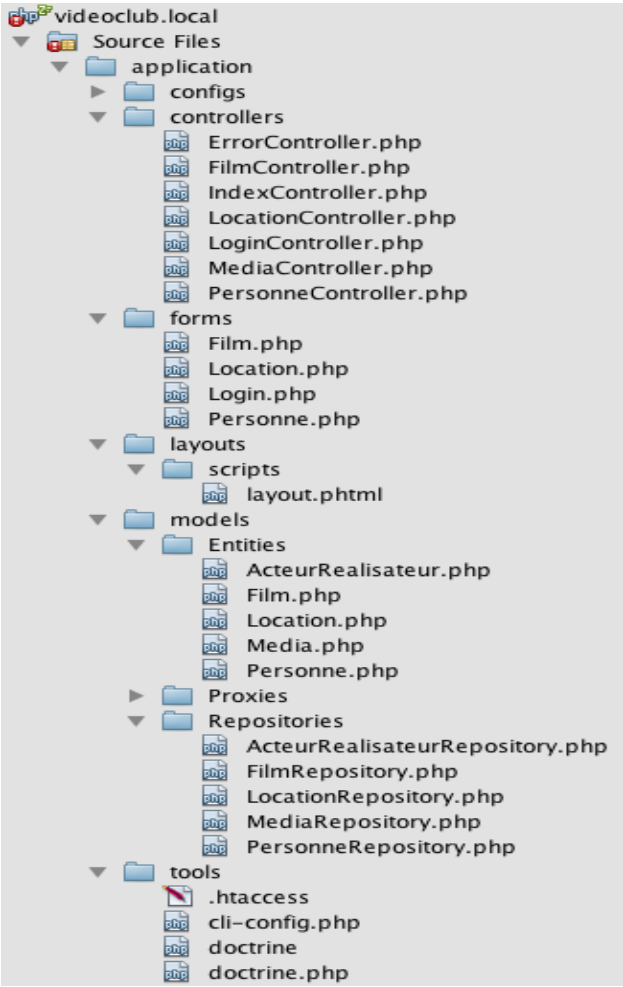
J'ai installé Doctrine (en V2.2). N'ayant jamais du l'installer avec symfony2, l'installation n'as finalement pas été si difficile grâce un bon tuto trouvé sur le net.

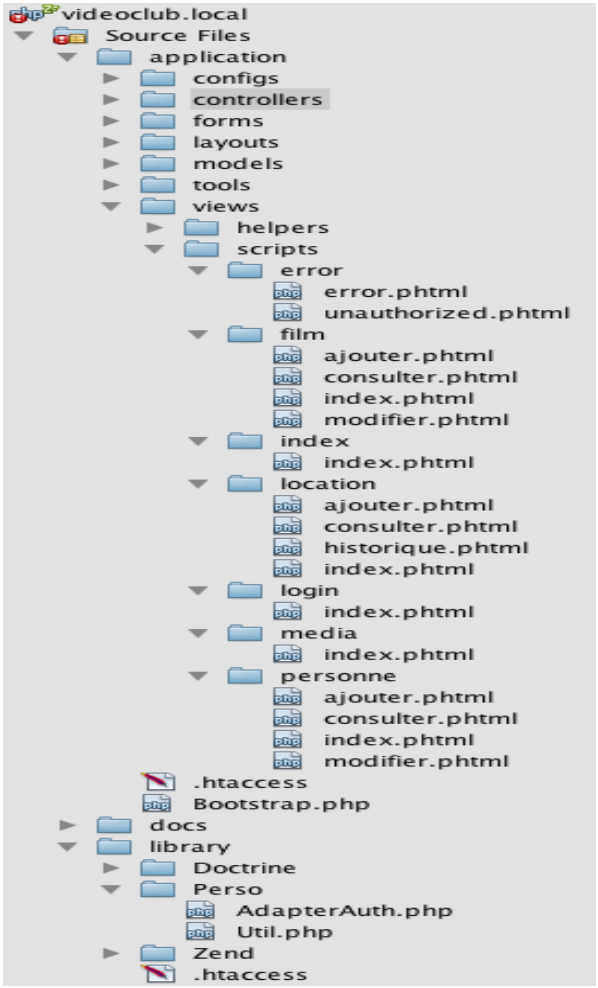
En écrivant les annotations de mes entités, j'ai fait attention de respecter au maximum votre modèle sql pour la BDD. J'ai généré mon script SQL de création de la base à partir du modèle de mes entités grâce a la commande **#doctrine orm:shema:update --dump-sql**.

Coté client :

- Javascript : JQuery, JQuery-UI (pour DatePicker et Dialog), Datatable.
- CSS

## Contenu des répertoires :




## Spécificités :

- J'ai surchargé le constructeur des formulaires Film et Personne pour pouvoir les initialiser en Création ou en Modification. Ainsi ils ne retournent pas les mêmes éléments en fonction de comment il sont initialisés.
- J'ai rajouté une action dans le controleur Erreur pour traiter les accès à des pages non autorisés.
- J'ai créé un layout utilisé par toutes les pages.
- Toutes les actions du controller Media retourne des données au format JSON (pas de vues), c'est que pour des appels Ajax.
- Les entités et les repositories sont dans le dossier application/models
- Un dossier Perso dans le dossier library contient deux classes :
  - AdapteurAuth : Mon adapter perso pour l'authentification.
  - Util : mets à disposition des méthodes static utilisées partout dans l'appli.
  - Les namespaces me manquent …
- Un controller login pour la connexion et la déconnexion.

- C’est l’entité Personne qui gère le cryptage du mot de passe. Le password de l’utilisateur est crypté au niveau du setter “setPassword” de l’entité Personne. De même, la fonction qui teste le mot de passe de la personne est “verifierPassword()” dans l’entité Personne.
- Un dossier tool qui contient les classes Doctrine pour effectuer des actions en ligne de commandes (le fichier de conf de la BDD est config-cli.php).

## Divers :

- Le code source est dans le dossier **videoclub** du Zip
- Le script de creation de la BDD : **create_insert.sql** dans le Zip et dans le dossier docs à la racine du projet.
- Le mot de passe de tous les utilisateurs est “**azerty**” dans mon script SQL.
- Un compte admin est “**admin@video.club**” dans mon script SQL.
- application.ini contient les paramètres Doctrine pour la connexion à la BDD.

## Pas eu le temps de faire :

- ActeurRealisateurs : Pas de gestion, et la selection des acteurs réalisateur dans mes vues est sommaire : une liste dans un popup.
- Utilisateurs : Ils ne peuvent pas changer leur mot de passe. Un employé peut quand même le leur réinitialiser.
- DataTables : Je les ai utilisés dans mes vues pour profiter des champs de recherche et de tri. Je sais que si le nombre d’enregistrement explose, la vue aussi. Mais je n’avais pas le temps de faire une interface de recherche qui limite le nombre d’occurrence envoyé a la vue.
- Dojo : Dommage que votre cours sur ce sujet arrive si tard. Je ne connaissais pas, ca à l’air plutôt sympa et je me serais bien fait les dents dessus !
- Pas de Log
- WebService AlloCine : Etudié mais pas mis en place.