

## PROJET – PHP / MYSQL

### RÉALISATION D'UN SITE DE PARTAGE DE RECETTES DE CUISINE

---



#### Objectifs

---

**L'objectif de ce TP est de mettre en place l'architecture nécessaire pour la mise en œuvre des espaces membre et administration.**

**L'espace *MEMBRE* permet à un utilisateur connecté de :**

Créer une recette et la proposer à la publication ;

Tant que la recette associée n'est pas validée, gérer les ingrédients et unités créés pour celle-ci ;

Gérer son profil (Nom, prénom, mail, login, mot de passe, etc.).

**L'espace *ADMINISTRATION* permet de :**

Gérer les recettes, leurs ingrédients et unités, proposés par les membres (rôle de modération) ;

Gérer les ingrédients et les unités existants ;

Gérer les comptes utilisateur (les membres et les autres administrateurs).

Afin de garantir que seuls les membres auront accès aux fonctionnalités qui leur sont dédiées, vous allez modifier le **CORE** de codeIgniter. Pour cela, vous devez mettre en place l'héritage tel qu'il vous est présenté sur la Figure 1. Chaque package de la figure représente en fait un répertoire de CodeIgniter.

Pour bien comprendre, il vous est conseillé d'aller lire la documentation relative à la modification du CORE de CodeIgniter :  
[http://codeigniter.com/user\\_guide/general/core\\_classes.html](http://codeigniter.com/user_guide/general/core_classes.html)

#### Prérequis

---

Avoir fini le TP4

## Mise en place de l'architecture

Vous allez créer une classe *MY\_CONTROLLER* qui hérite de *CI\_CONTROLLER*. Cette classe possède un attribut *user* qui est un objet (ou tableau associatif – à vous de voir) qui contient les informations sur l'utilisateur connecté.

Cette classe implémente également les quatre méthodes suivantes :

### **isLogOn()**

Renvoie TRUE si un utilisateur est connecté, FALSE sinon.

### **isAdmin()**

Renvoie TRUE si un utilisateur est connecté et que c'est un administrateur, FALSE sinon.

### **connexion()**

Cette fonction s'occupe de la connexion de l'utilisateur i.e. crée en session ([http://codeigniter.com/user\\_guide/libraries/sessions.html](http://codeigniter.com/user_guide/libraries/sessions.html)) une variable permettant d'authentifier un utilisateur et renseigne l'attribut *user*.

### **deconnexion()**

Cette fonction détruit les informations de connexion, de la session et de la classe.

Le constructeur de cette classe aura pour rôle de récupérer les informations relatives à un utilisateur si celui-ci est connecté. Si aucun utilisateur est connecté, vous pouvez affecter la valeur NULL, ou FALSE à *user*, en fonction de vos choix précédents.

Le constructeur de la classe *MY\_Membre\_Controller* aura pour rôle de vérifier que l'on a bien un utilisateur authentifié. Dans le cas contraire, il opérera une redirection vers le contrôleur du FrontOffice (*Home* dans le cas de ce schéma).

Le constructeur de la classe *MY\_Admin\_Controller* aura pour rôle de vérifier que l'on a bien un utilisateur authentifié et que celui-ci est bien un administrateur. Dans le cas contraire, il opérera une redirection vers le contrôleur du FrontOffice également.

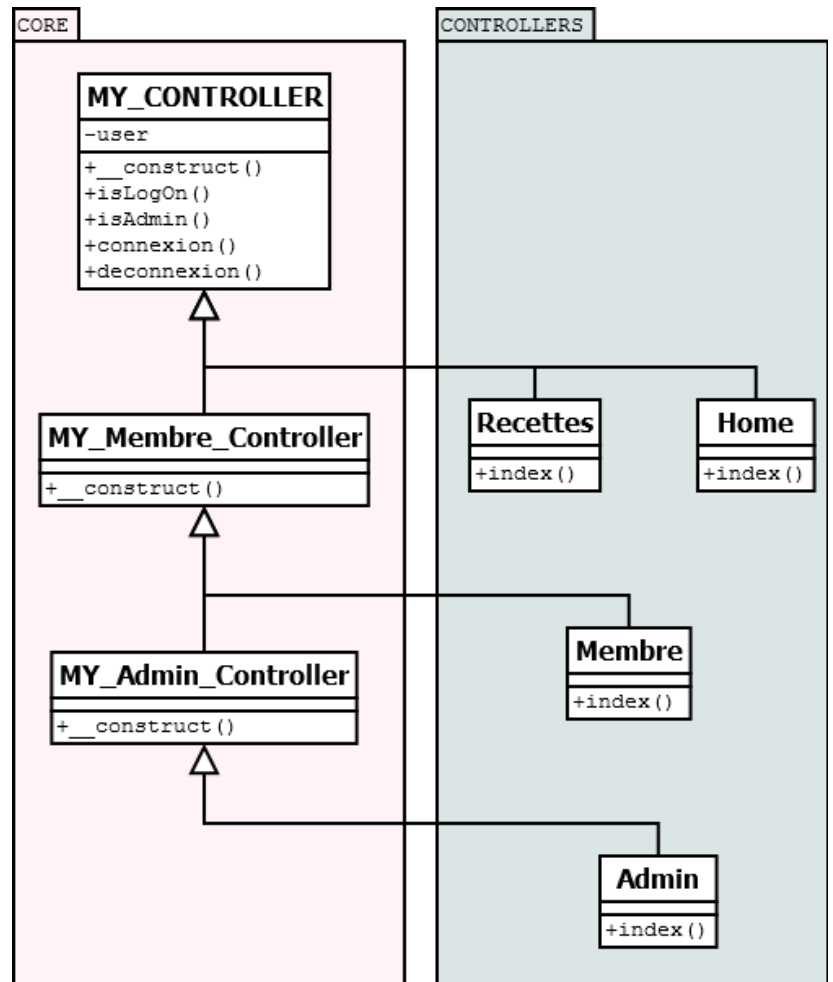


Figure 1

Dès lors, chaque fois que vous voudrez mettre en place un contrôleur pour votre site et que celui-ci regroupe des fonctionnalités réservées aux membres, il vous suffira de le faire hériter de *MY\_Membre\_Controller*. Si les fonctionnalités sont réservées aux administrateurs du site, alors, votre contrôleur devra hériter de *MY\_Admin\_Controller*. Cette approche offre beaucoup de souplesse et vous permet surtout d'éviter d'oublier de protéger une fonctionnalité.

## Espace Membre et Administration

---

Maintenant que l'architecture est en place, il ne vous reste plus qu'à créer pour chaque contrôleur les fonctionnalités nécessaires (création de recettes, ajout d'ingrédients, modération, etc.) ainsi que les vues qui vont avec. Cette étape n'est pas différente de ce que vous avez fait lors du TP précédent, si ce n'est que vous aurez besoin d'avoir recours aux formulaires. Il pourrait alors être utile de regarder ceci :

[http://codeigniter.com/user\\_guide/libraries/input.html](http://codeigniter.com/user_guide/libraries/input.html)

[http://codeigniter.com/user\\_guide/libraries/form\\_validation.html](http://codeigniter.com/user_guide/libraries/form_validation.html)

[http://codeigniter.com/user\\_guide/libraries/file\\_uploading.html](http://codeigniter.com/user_guide/libraries/file_uploading.html)