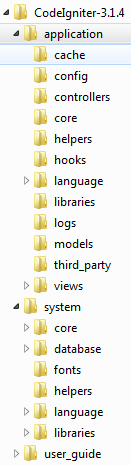
CodeIgniter

CodeIgniter est un Framework se basant sur le modèle MVC. Il dispose de nombreuse librairie qui permette son utilisation.

L’arborescence de CodeIgniter se profile comme suit :

Le dossier « config » contient les toutes les configurations (database, routes, autoload, …). Les modifications de configuration doivent se faire dans ce dossier.  
Dans le dossier « controllers » se trouve tous les fichiers de contrôle du MVC.   
« models » contiendra tous les fichiers modèle de l’MVC.  
« Views » contiendra les fichiers vue de l’MVC.  
Le dossier « language » contiendra les versions de votre site. Utiliser uniquement si l’on souhaite de créer un site multilingue.

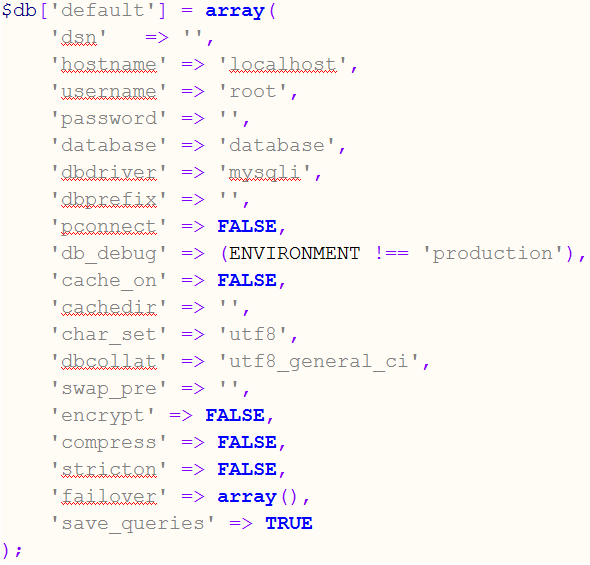
Ce petit guide n’as d’utilité qu’à donner les bases du développement avec CodeIgniter. Pour de plus amples information sur une fonctionnalité, rendez-vous sur la documentation [ici](https://codeigniter.com/user_guide/general/index.html).

Table des matières

[Syntaxe View : 3](#_Toc480363218)

[Syntaxe Model : 4](#_Toc480363219)

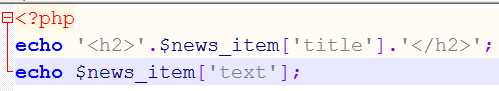
[Syntaxe controllers : 5](#_Toc480363220)

Exemple du fichier « database.php »  dans le dossier « config ». Le fichier « database » contient les informations relatives à la connexion à la base de données : 

Les exemples qui suivent sont tirés du tutoriel de CodeIgniter que l’on peut retrouver [ici](https://codeigniter.com/user_guide/tutorial/index.html)[[1]](#footnote-1).

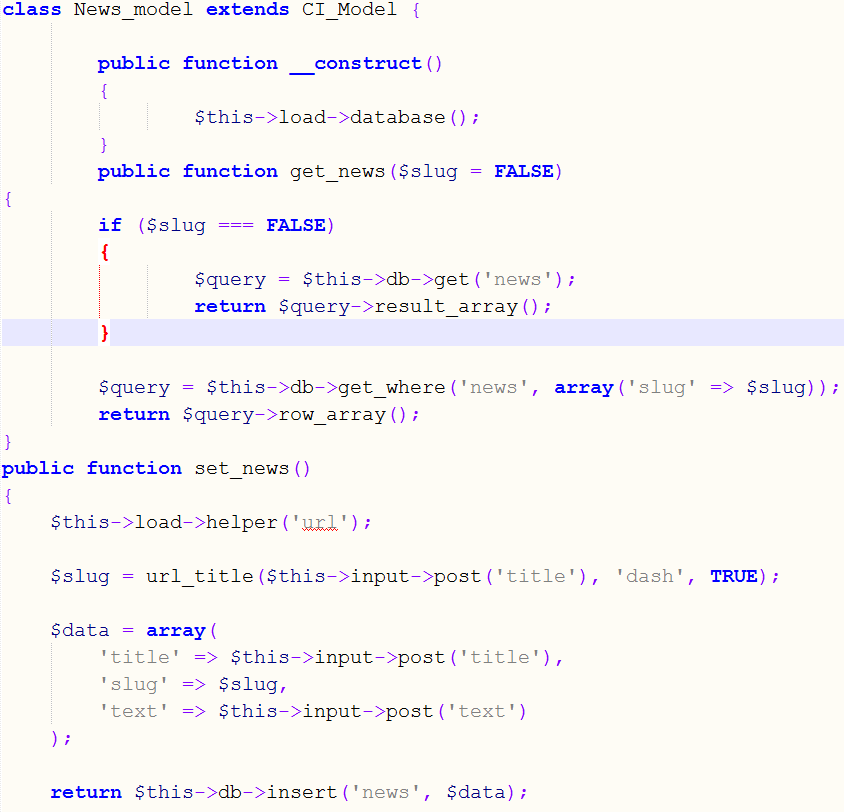
# Syntaxe View :

Le view contiendra essentiellement de l’html. Il peut également contenir du PHP lorsque l’on doit récupérer des variables / utiliser des variables dans une boucle, conditions ou autre. Le view sera ensuite insérer dans le Controller (CF Syntaxe controllers :).  
Exemple de la view « news » :



# Syntaxe Model :

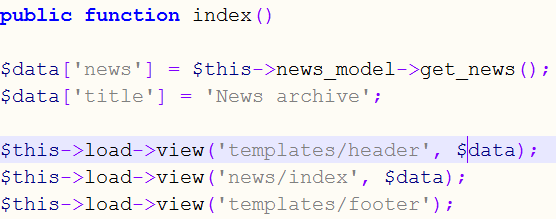
Chaque model sera une classe étendue de CI\_Model  
Dans l’exemple, une requête est faite à une base de données. Cette requête est placée dans un constructeur. On retrouve ensuite les fonctions « get\_news » et « set\_news ». Tout sera ensuite utilisé par le Controller (CF Syntaxe controllers :)



Extends CI\_Model permet d’étendre CI\_Model et de charger les Bibliothèques de « database ». Ceci permet de rendre la classe database disponible. La syntaxe $this->load->database() ; permet l’utilisation de la fonction database et d’utiliser sa fonction database qui se chargera de la connexion avec la base de données.

Note : Il est important de configurer le mot de passe, le nom de la base de données ainsi que le nom utilisateur. Le fichier qui doit être modifié se trouve dans le dossier : CodeIgniter-3.1.4\application\config\database.php

# Syntaxe controllers :

Chaque Controller sera une classe.  
Chaque page sur le site web se traduira par étant une fonction de la classe. Dans l’exemple j’ai une page qui se nomme ‘index’ et une page qui se nomme ‘view’. Dans mon Controller j’aurais donc une fonction public ‘index()’ et une autre ‘view()’. La fonction ‘index()’ se présente comme suit :

$this->news\_model->get\_news() : Permet d’utiliser la fonction get\_news() se trouvant dans le model. Les variables se voient donc attribuée des informations se trouvant dans la base de données. Ces données seront ensuite utilisées par le view.

$this->load->view(‘templates/header’, $data) ; Cette ligne permet de charger le header et d’y envoyer les informations comprises dans $data. La ligne suivante fait pareille. La dernière ne fait que charger la page. Cela revient à faire un require\_once() en y envoyant des informations.

1. https://codeigniter.com/user\_guide/tutorial/index.html [↑](#footnote-ref-1)