

## Projet CodeIgniter (niveau débutants/intermédiaires) [[www.bleekom.org](http://www.bleekom.org)]

Au cours de cet article nous allons voir comment utiliser le framework CodeIgniter.

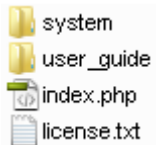
Ce framework évolue constamment et il est compatible PHP4/PHP5. Un grand nombre de bibliothèques et helpers en tout genre permettent aussi de ne pas avoir à tout réinventer et de gagner du temps dans des développements structurés. Plus facile à prendre en main que Symfony ou Zend, il suscite un engouement croissant parmi la communauté PHP.

Installer une solution APACHE/MySQL/PHP : WAMP ou XAMPP.

Télécharger la dernière version de CodeIgniter : (<http://codeigniter.com/downloads/>) actuellement 1.7.1.

Sous la racine du serveur web (htdocs pour XAMPP), créer un nouveau répertoire (CI) et y décompresser l'archive de CodeIgniter :

### C:\xampp\htdocs\CI



Nous avons sous CI, 2 répertoires :

- **system** qui définit la structure de l'application
- **user\_guide** qui est le guide utilisateur de CodeIgniter.

Se trouve aussi le fichier index.php qui comprend l'assignation des différents paramètres par défaut, et qui indique à CodeIgniter où trouver les fichiers d'une application

```
$system_folder = "system";
```

```
$application_folder = "application";
```

```
le controller principal : require_once BASEPATH.'codeigniter/CodeIgniter'.EXT;
                        BASEPATH - Le chemin serveur complet vers le répertoire "system"
                        EXT - extension du fichier. Typiquement ".php"
```

Lorsque que l'on va taper l'adresse : <http://localhost/CI> , le fichier index.php va donc aller chercher le fichier CodeIgniter.php dans le répertoire system/codeigniter.

```
$CFG =& load_class('Config');
$URI =& load_class('URI');
$RTR =& load_class('Router');
$OUT =& load_class('Output');
```

Les fichiers des classes se trouvent dans le répertoire « libraries » de CI\system\.

La classe 'Router' va définir le fichier **Route.php** @include (APPPATH.'config/routes'.EXT) ; à charger, qui se trouve sous CI\system\config\.

Ce dernier va définir le contrôleur à utiliser :

```
$route['default_controller'] = "welcome";
```

Sous CI\system\application\controllers, nous retrouvons le fichier welcome.php qui en tant qu'extension de l'objet 'Controller' comporte 2 méthodes :

```
class Welcome extends Controller {  
  
    function Welcome()  
    {  
        parent::Controller();  
    }  
    function index()  
    {  
        $this->load->view('welcome_message');  
    }  
}
```

La première méthode est le constructeur de la classe Welcome et reprend le constructeur de l'objet parent Controller dont la classe Welcome hérite.

La deuxième méthode 'index' charge la vue 'welcome\_message'.

Cette dernière se trouve dans CI\system\application\views et se nomme welcome\_message.php.

Dans ce dernier fichier, nous retrouvons la structure d'une page HTML affichant un message de bienvenue

## Welcome to CodeIgniter!

---

The page you are looking at is being generated dynamically by CodeIgniter.

If you would like to edit this page you'll find it located at:

```
system/application/views/welcome_message.php
```

The corresponding controller for this page is found at:

```
system/application/controllers/welcome.php
```

If you are exploring CodeIgniter for the very first time, you should start by reading the [User Guide](#).

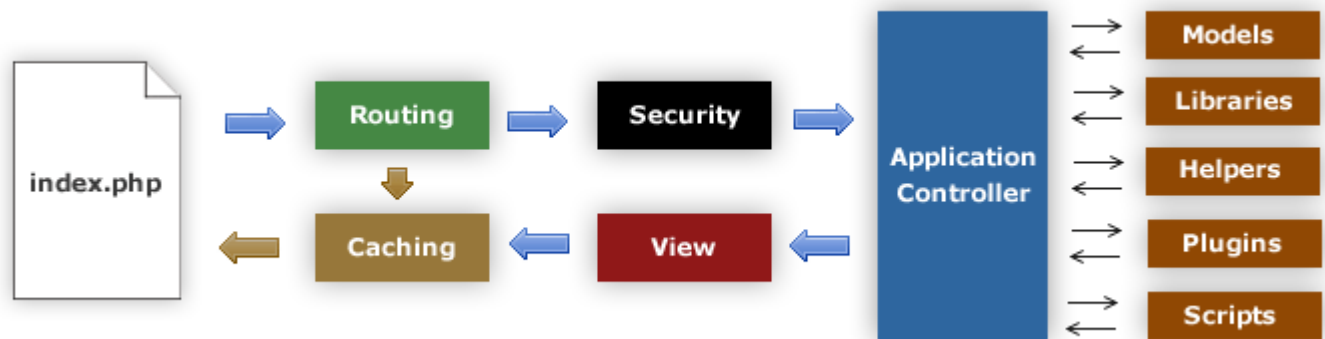
Page sent in 0.0616 seconds

---

Nous avons donc vu dans cette première partie un extrait du processus MVC utilisé par CodeIgniter.

index.php sert de contrôleur frontal initialisant les ressources de base nécessaires à l'utilisation du Framework.

La partie Router examine la requête HTTP et détermine ce qu'il doit en faire, en l'occurrence charger le contrôleur 'welcome', lequel charge la vue 'welcome\_message.php'.



Comme nous le voyons sur le graphique ci-dessus, la puissance de CodeIgniter ne s'arrête pas là.

En effet, via le contrôleur, un ensemble de fichiers de configuration permettent de charger la base de données utilisable par une application, les modèles et routines d'accès aux données ainsi qu'un ensemble de bibliothèques (classes), de helpers, de plugins et de scripts facilitant encore plus le développement rapide d'application.

Nous allons voir comment créer une petite application basique d'une boutique en ligne listant sur une page d'accueil les produits en magasin et présents en base de données.

### Utiliser une Base de données :

Créer via PhpMyadmin, une base 'ci\_shop'.

CodeIgniter possède un fichier de configuration qui permet de stocker les paramètres de connexion à votre base de données ((username, password, database name, etc.). Le fichier de configuration se trouve sous :

`CI/system/application/config/`

`database.php`

Les paramètres sont stockés dans un tableau multi-dimensionnel sous cette forme :

```
$active_group = "default";
$active_record = TRUE;

$db['default']['hostname'] = "localhost";
$db['default']['username'] = "root";
$db['default']['password'] = "";
$db['default']['database'] = "ci_shop";
$db['default']['dbdriver'] = "mysql";
$db['default']['dbprefix'] = "";
$db['default']['pconnect'] = TRUE;
$db['default']['db_debug'] = TRUE;
$db['default']['cache_on'] = FALSE;
$db['default']['cachedir'] = "";
$db['default']['char_set'] = "utf8";
$db['default']['dbcollat'] = "utf8_general_ci";
```

Créer ensuite via PhpMyadmin une table ci\_articles :

```
CREATE TABLE IF NOT EXISTS `ci_articles` (  
  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `court_desc` varchar(255) COLLATE latin1_general_cs NOT NULL,  
  `long_desc` varchar(500) COLLATE latin1_general_cs NOT NULL,  
  `prix_ht` decimal(10,0) NOT NULL,  
  `img_thumb` varchar(255) COLLATE latin1_general_cs NOT NULL,  
  `img_std` varchar(255) COLLATE latin1_general_cs NOT NULL,  
  PRIMARY KEY (`id`)  
  
) ENGINE=MyISAM DEFAULT CHARSET=latin1 COLLATE=latin1_general_cs AUTO_INCREMENT=3 ;  
  
--  
-- Contenu de la table `ci_articles`  
--  
  
INSERT INTO `ci_articles` (`id`, `court_desc`, `long_desc`, `prix_ht`, `img_thumb`, `img_std`) VALUES  
(1, 'The pc', 'Le Pc le plus vendu dans le monde', 399, 'images\theipc.jpg', 'images\theipc_s.jpg'),  
(2, 'The phone', 'Le téléphone intuitif sans limites', 499, 'images\thephone.jpg', 'images\thephone_s.jpg');
```

Nous allons ensuite créer un modèle d'accès aux données de la table ci\_articles :

Sous C:\system\application\models, créer le fichier marticles.php et entrer le code suivant dedans :

```
<?php class MArticles extends Model {  
  
  //constructeur  
  function MArticles() {  
    parent::Model();  
  }  
  //récupère tous les articles de la table ci_articles  
  function getAllArticles() {  
    $data = array();  
    $Q = $this->db->get('ci_articles');  
    if ($Q->num_rows()>0) {  
      foreach ($Q->result_array() as $row) {  
        $data[]=$row;  
      }  
    }  
    $Q->free_result();  
    return $data;  
  }  
  //  
} //end class  
?>
```

Modifions ensuite le fichier autoload.php sous C:\system\application\config

```
$autoload['libraries'] = array('database');
```

```
$autoload['model'] = array('MArticles');
```

Modifier ensuite le fichier 'welcome.php' sous C:\system\application\controllers de la façon suivante :

```
<?php

class Welcome extends Controller {

    function Welcome ()
    {
        parent::Controller();
    }
    function index()
    {
        //$this->load->view('welcome_message');
        //appel à la méthode 'getAllArticles' du Model 'MArticles'
        $data['articles']= $this->MArticles->getAllArticles();
        //charge les données à transférer à la vue
        $this->load->vars($data);

        //charge la vue 'list_articles'
        $this->load->view('list_articles');
    }
}

/* End of file welcome.php */
/* Location: ./system/application/controllers/welcome.php */
```

Puis sous C:\system\application\views, créer la vue nommée **list\_articles.php**

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
<meta http-equiv="content-type" content="text/html; charset=UTF-8" />

<title>Liste des articles</title>

<style type="text/css">
body { background-color: #fff;
margin: 10px;
font-family: Lucida Grande, Verdana, Sans-serif;
font-size: 12px;
color: #4F5155;
}
</style>
</head>

<body>
<?php

echo "<b>Liste des articles :</b><br/>";
```

```
foreach ($articles as $key => $list){  
  
    echo "<br/>".$list['id'].  
    " <br/>- ".utf8_decode($list['court_desc']).  
    " <br/>- ".utf8_decode($list['long_desc']).  
    " <br/>- ".$list['prix_ht']."<br/>";  
}  
?>  
  
<p><br />Page affich&eacute;e en {elapsed_time} secondes</p>  
  
</body>  
</html>
```

Lancer <http://localhost/CI> ou en réel sur mon site <http://www.bleekom.org/ci>

Vous devriez voir apparaître la liste de tous les articles de la table ci\_articles :

**Liste des articles :**

1  
- The pc  
- Le Pc le plus vendu dans le monde  
- 399

2  
- The phone  
- Le téléphone intuitif sans limites  
- 499

Page affichée en 0.1193 secondes