

Utilisation du Framework CodeIgniter

CodeIgniter est un framework simple qui permet de concevoir une application web de type MVC. Vous utiliserez le framework dans le cadre de la réalisation d'une application web simplifiée de vente de livres en ligne.

I) Préambule

a) mise en place

- importez dans *mysql* les 2 tables dont le code *sql* vous est fourni dans *sql.tar.gz* sur *madoc*.
- récupérez l'archive *codeIgniter-3.1.6.tar.gz* sur *madoc* et décompressez la dans votre *public_html*.

b) paramétrage

Nous allons maintenant configurer quelques fichiers pour paramétrer le framework.

- dans *application/config/config.php*, ligne 26, renseignez la variable `$config['base_url']` avec l'URL pour accéder au répertoire *codeIgniter-3.1.6* qui vous concerne (par exemple : `http://infoweb/~votre_identifiant_etudiant/codeIgniter-3.1.6`). Ce sera maintenant votre URL de base.
- dans *application/config/database.php*, ligne 76, modifiez les données de connexion au SGBD *mysql* dans le tableau `$db['default']`. Il suffit de remplacer les trois X par vos propres identifiants de connexion.
- dans *application/config/autoload.php*, vérifiez que vous avez `$autoload['libraries'] = array('database','session');`
Ceci permet d'avoir accès aux classes du modèle utilisant une base de données et de pouvoir utiliser les sessions . Vous n'aurez pas besoin, lors de l'emploi des sessions, d'utiliser `session_start()`. Le framework s'en chargera de manière transparente pour vous.

II) Les contrôleurs et les vues

Dans le répertoire *application*, vous avez 3 répertoires : *controllers*, *views*, *models* qui respectivement contiennent les fichiers relatifs aux contrôleurs, vues et modèles de l'application.

La première page du site vous est fournie. Pour charger cette page dans un navigateur saisissez l'adresse suivante:
`http://infoweb/~votre_identifiant_etudiant/codeIgniter-3.1.6/index.php/ControleurAccueil/accueil`. L'item "connexion" n'est pas fonctionnel.

Les URL sont conçues avec une partie fixe: `http://infoweb/~votre_identifiant_etudiant/codeIgniter-3.1.6/index.php` et une partie variable qui est le contrôleur appelé (ici *ControleurAccueil*) suivi de la méthode du contrôleur qui est sollicitée. On verra plus tard qu'il est aussi possible de passer des données à cette méthode.

Ouvrez le fichier *ControleurAccueil.php* qui est situé dans le répertoire *controllers* pour visualiser ce fait.

La méthode `accueil()` du contrôleur *ControleurAccueil* est chargée de lancer une vue qui se nomme *accueil*. Dans le répertoire *views* vous avez le script php concerné qui se nomme *accueil.php*.

Noter qu'il faudra développer une page php par vue différente de votre site.

III) Travail à réaliser

1) Préambule

La maquette du site que vous allez réaliser a été conçue avec un rendu minimaliste (mais vous pouvez retravailler la charte graphique et les feuilles de style à votre guise). Les feuilles de style sont dans le répertoire *assets*.

La partie modèle vous est fournie. Les fichiers concernés se trouvent dans le répertoire *dao* situé dans le répertoire *models*. Quatre méthodes dont vous aurez besoin sont développées dans la classe *Dao*. Si vous regardez d'un peu plus près leur code, des objets et leurs méthodes propres à *codeIgniter* sont utilisés. Si vous voulez des détails : <http://www.codeigniter.com/userguide3/database/results.html>.

Dans la table *identification*, les logins et mots de passe sont identiques (*toto* et *titi*).

2) Travail à réaliser

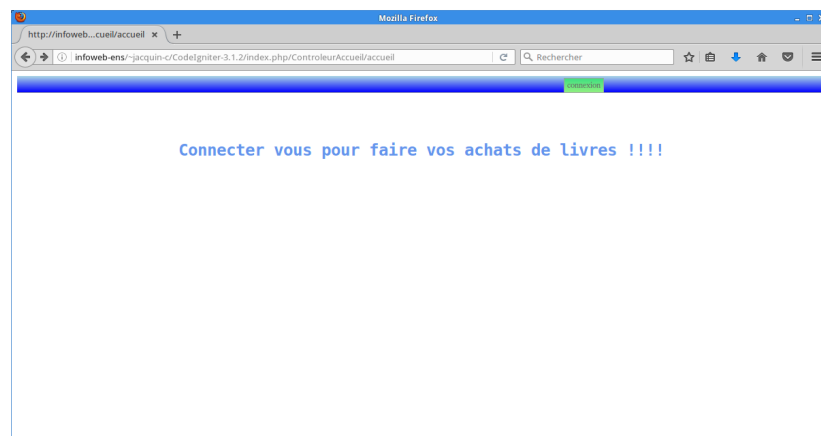
a) La classe *Livre* vous est fournie dans *application/libraries*. Vous avez à développer les classes *Article* et *Panier* dans des fichiers séparés que vous stockerez dans le répertoire *application/libraries*. Un objet de type *Livre* permet de stocker les données contenues dans la table *livres* pour un livre donné. Un objet de type *Article* disposera d'un attribut de type *Livre*, d'un attribut de type *entier* qui représentera le nombre d'exemplaire commandé d'un livre et d'un autre attribut de type *flottant* qui représentera le montant relatif à la commande des exemplaires du livre. Un objet de type *Panier* permettra de stocker des objets de type *Article*. Il comportera une méthode d'ajout d'article dans le panier (attention au cas où un article est déjà dans le panier) et d'une méthode permettant de calculer la somme globale de la commande.

b) la page d'accueil => vue A

Dans un navigateur saisissez l'URL :

http://infoweb/~votre_identifiant_etudiant/codeIgniter-3.1.6/index.php/ControleurAccueil/accueil

La vue A s'affiche. L'item "connexion" est lié au contrôleur qui lancera la vue B. Regarder comment ceci s'écrit dans *application/views/bandeau/bandeauAccueil.php* (dans le lien hypertexte).



c) la page d'authentification => vue B

- développez le contrôleur *ControleurConnexion* et sa méthode *voir()* qui lance la vue B (la méthode *voir()* est invoquée quand l'utilisateur clique sur l'item "connexion" de la vue A).

Le bandeau de la vue B est dans *application/views/bandeau/bandeauPendantConnexion.php*



- développez la méthode *connecte()* dans le contrôleur *ControleurConnexion* qui a pour charge de tester si le couple login et mot de passe entré par l'utilisateur existe dans la base de donnée. Si c'est le cas, elle permet l'affichage de la vue C, sinon elle réaffiche la vue A. Vous aurez besoin de la classe *Dao* qui fait partie du modèle. Voici les instructions pour charger une classe du modèle (en général ce chargement est effectué dans le constructeur de la classe qui nécessite le modèle)

```

public function __construct(){
parent::__construct();
// charge la classe Dao qui fait partie du modèle
// elle se trouve dans models/dao
$this->load->model('dao/Dao');
}

```

Ensuite, par exemple pour utiliser la méthode *methode()* de la classe *Dao* => *\$this->Dao->methode()* ;
Remarquer qu'on n'instancie pas d'objets liés au modèle, on a directement accès à *\$this->Dao*

d) La page après authentification => vue C (bandeau : *application/views/bandeau/bandeauApresConnexion.php*)



- développez le contrôleur *ContrôleurLivre*. Sa méthode *voir()* sera invoquée lorsque l'utilisateur cliquera sur l'item "afficher livres". Elle permettra l'affichage de la vue D.
Pour développer ce contrôleur, il faut charger la classe *Livre* stockée dans le répertoire *libraries* dans son constructeur.

```

public function __construct(){
parent::__construct();
...
// chargement de la classe Livre
$this->load->library('Livre');
...}

```

Attention, si vous voulez utiliser dans votre code un objet de type *Livre*, il faudra auparavant l'instancier.

Pour afficher la vue D, il faudra que la méthode *voir()* du contrôleur *ContrôleurLivre* envoie à la vue les livres présents dans la base. Dans ce but, un contrôleur passe des données à une vue par le biais d'un tableau associatif.

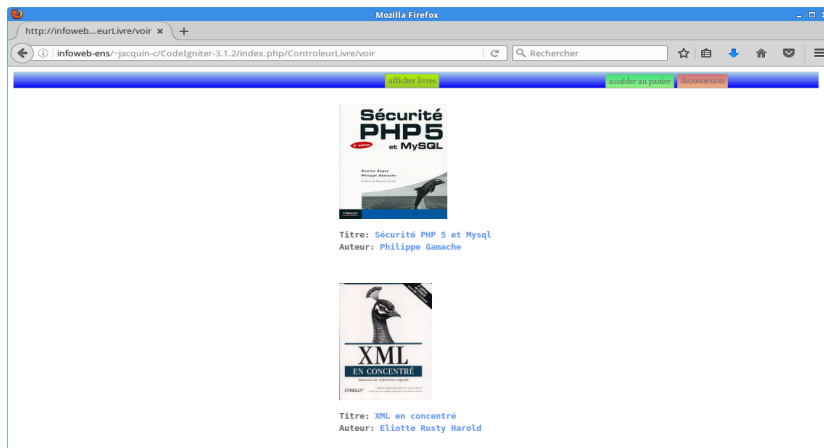
```

class ContrôleurPersonne extends CI_Controller {
    public function identité() {
        $data = array('nom' => 'Pierre');
        $this->load->view('maVue', $data);
    }
}

```

Ensuite, on accède dans *maVue.php* aux données communiquées par le contrôleur via *\$nom*. Il est à noter qu'on peut stocker dans le tableau associatif des données de divers types (tableau, objets...). Le tableau associatif peut aussi contenir plusieurs éléments.

e) La page qui affiche la liste des livres disponibles => vue D



Rajoutez la méthode *afficherLivre()* dans le contrôleur *ControleurLivre* qui lorsque l'utilisateur cliquera sur l'image d'un livre (par exemple : *sécurité PHP 5 et MySQL*) dans la page D, permettra d'afficher la vue E.

Vous allez avoir besoin à ce niveau de passer une donnée en URL longue. Voici comment cela se passe sous CodeIgniter avec un exemple :

Si la méthode appelée du contrôleur a des paramètres, les valeurs passées lors de l'appel de la méthode s'ajoute au bout de l'URL de cette manière (ex avec 2 paramètres)

.../ControleurAccueil/maMethode/valeur1/valeur2/

Si vous avez besoin d'utiliser des sessions voir dans la partie IV pour les explications.

f) La vue qui affiche le livre choisi et permet de saisir le nombre d'exemplaire voulu => vue E



Développez la méthode *stocker()* dans le contrôleur *ControleurPanier* qui lorsque l'utilisateur cliquera sur "ajouter panier" permettra d'afficher la vue F.

g) la vue qui permet d'attester que l'ajout a bien été effectué => vue F



Dans le contrôleur *ControleurPanier*, développez la méthode *voir()*. Cette méthode est invoquée quand l'utilisateur cliquera sur l'item "accéder au panier" du bandeau. Elle permet de lancer la vue G.

h) la vue qui permet d'afficher le contenu du panier => vue G

Développez la méthode *deconnecte()* dans le contrôleur *ControleurConnexion* qui permet de déconnecter l'utilisateur du site.



IV) Utilisation des sessions avec CodeIgniter

Les sessions s'utilisent comme en php de base (variable super globale `$_SESSION[]`). Par contre, les objets doivent être sérialisés au moment de leur mise en session et désérialisés quand un script désire les utiliser (fonction *serialize()* et *unserialize()*). La gestion des sessions a été activée dans le fichier de configuration *config.php*, donc elles peuvent être utilisées sans autre ajout dans le code (pas de *session_start()*)

Pour détruire une variable de session par exemple nommée "panier" => `unset($_SESSION['panier'])`

Des liens pour vous aider:

- un tutoriel simple: <http://www.tutora.fr/apprendre-codeigniter>
- la documentation : http://www.codeigniter.com/user_guide/