

Programmation orientée Objet

Architecture MVC : Modèle Vue Controlleur

L'architecture MVC va permettre de séparer les parties du site en plusieurs morceaux

Elle se décompose en trois parties :

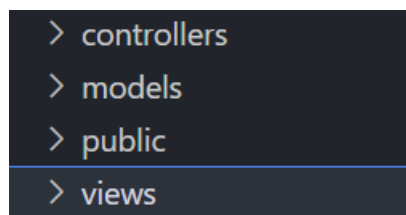
- Modèle : Les données
- Vue : L'affichage
- Contrôleur : traitement

Cette architecture permettra une meilleure sécurisation du code, une maintenabilité plus simple et assurera la cohérence globale du site.

Le principal inconvénient de ce type de structure est que la mise en place peut paraître lourde, surtout pour des débutants.

Au final, le site n'aura plus qu'une seule page : la page index.php qui fera office de « routeur ».

L'architecture



Je vous laisse créer les dossiers manquant, et vous allez en plus ajouter dans le dossier views le fichier accueil.view.php

Le routeur

Maintenant que nous faisons notre architecture MVC nous allons utiliser notre index comme routeur, c'est-à-dire que tous les utilisateurs arriveront sur notre index et c'est celui-ci qui s'occupera d'appeler nos contrôleurs.

Il va falloir :

- Créer un fichier accueil.view.php dans le dossier views
- Copier l'intégralité de votre index dedans, votre index se retrouvera vide.

Dans votre index :

- Il va falloir rediriger l'utilisateur vers l'accueil si GET['page'] est vide, en effet nous allons utiliser la méthode GET pour notre routage.
- Sinon on va faire un switch sur notre GET et si c'est l'accueil qui est demandé nous allons également afficher notre accueil. (**require de notre page accueil**)

HTACCESS

Nous commençons maintenant à avoir un semblant de routage fonctionnel, par contre quelque chose me chagrine. Les urls avec « ?page= » ne sont pas très esthétiques et en plus niveau référencement c'est pas top ...

Pas de panique ! Nous allons régler cela.

Vous allez créer un fichier .htaccess et y mettre ça dedans .

RewriteEngine On

RewriteCond %{REQUEST_FILENAME} !-f

RewriteCond %{REQUEST_FILENAME} !-d

RewriteRule ^(.*)\$ index.php?page=\$1

Cela nous permettra d'écrire les urls comme ceci « /accueil » plutôt que « ?page=accueil ». Il faut changer les url du template en conséquence.

Appliquons toutes la structure MVC :

- Mettre le template et la page des livres dans notre dossier views.
- Créez les routes correspondantes dans l'index.
- Changez les liens dans le header.

Dans notre books.view.php Il y a encore de la logique, nous utilisons le manager et faisons appel aux données des livres... Une vue ne doit contenir uniquement l'affichage et aucun traitement ou accès de données. Il va nous falloir un contrôleur.

Nous allons donc :

- Créer un controller : BookController.controller.php
- Créer une class BookController avec :
- Un attribut privé \$bookManager
- Un constructeur qui créera à l'instanciation notre manager et le stockera dans l'attribut bookManager. Le constructeur appellera également la méthode du manager pour générer tous les livres (loadingBooks() que vous devez créer).
- Une public function displayBooks qui :
- Créera notre liste de livre à l'aide du manager et appellera notre vue.
- Supprimez ces parties de la vue qui ne sont plus nécessaires.

Pour finir nous allons déplacer tout notre accès aux données dans le dossier models donc :

- Book.class
- BookManager
- Model