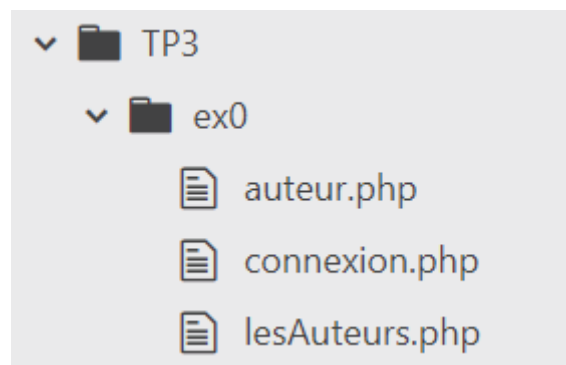


TP3 – introduction au pattern MVC

Exercice 1 – Un travail pas encore bien organisé

Vous allez travailler sur cette architecture de fichiers



- Le fichier `connexion.php` est déjà bien organisé, et est codé en PHP objet qui met en évidence une classe `Connexion`.
 - Le fichier `auteur.php` est aussi en PHP objet, c'est une bonne base. Il définit la classe `Auteur`.
 - Le fichier `lesAuteurs.php` est à repenser complètement. Il est simple, mais sa réorganisation va nous permettre de nous habituer au schéma MVC.
1. Créez un dossier `TP3/ex1`, puis un dossier `TP3/ex1/config` et déplacez-y le fichier de `connexion`.
 2. L'appel à la page `TP3/ex1/lesAuteurs.php` ne fonctionne pas. Expliquez pourquoi et corrigez.
 3. De même, créez un dossier `TP3/ex1/modele` et déplacez-y le fichier `auteur.php`.
 4. Arrangez-vous pour que la page `TP3/ex1/lesAuteurs.php` fonctionne avec cette nouvelle organisation.
 5. Il nous reste le fichier `lesAuteurs.php`. Dans ce fichier, identifiez clairement ce qui est du domaine de la vue et du contrôleur.

Exercice 2 – création de la vue et du routeur

1. Dupliquez `TP3/ex1` en `TP3/ex2` puis créez un répertoire `TP3/ex2/vue`. Placez-y une copie du fichier `lesAuteurs.php`. Comme il y a deux fichiers portant le même nom, renommez le fichier qui se trouve à la racine de `TP3/ex2` en `routeur.php`. On verra bientôt pourquoi ce nom.
2. Effacez de `lesAuteurs.php` tout ce qui est du domaine du contrôleur. Ce fichier vue ne produit que du code html, avec de l’affichage de variables déjà définies par ailleurs.
3. Effacez de `routeur.php` tout ce qui est du domaine de la vue, et ne laissez que ce qui est du domaine du contrôleur. Insérez un appel à la vue pour remplacer les lignes « vue » supprimées. Vérifiez que tout fonctionne !

Exercice 3 – création du contrôleur

Pour le moment, le fichier `routeur.php` est le seul qui fait des calculs par le biais du modèle (et donc de la base de données), et insère des vues. C’est donc lui le seul contrôleur pour le moment.

Mais bientôt il aura trop de choses à faire : des tâches sur les auteurs, mais aussi sur les livres. Il va donc avoir besoin de déléguer son travail à des « sous-contrôleurs ».

1. Dupliquez votre travail en `TP3/ex3`. Créez un dossier `TP3/ex3/contrroleur`, et créez un fichier `TP3/ex3/contrroleur/contrroleurAuteur.php`. Dans ce fichier, insérez les lignes du routeur qui sont vraiment spécifiques à la notion d’auteur :
 - a. L’insertion du modèle auteur
 - b. La récupération du tableau des auteurs
 - c. L’insertion de la vue des auteurs
2. Ne gardez dans `routeur.php` que :
 - a. L’insertion du fichier de connexion
 - b. L’appel de la méthode de connexion
 - c. L’insertion du contrôleur précédent
3. Vérifiez que tout fonctionne !

Exercice 4 – version objet du contrôleur

Vous allez maintenant produire une version objet du fichier `controleurAuteur.php`. Commencez par dupliquer votre travail en `TP3/ex4`.

1. Créez dans ce fichier une classe `ControleurAuteur`. Cette classe ne sera pas en fait destinée à être instanciée (pas d'appel à un constructeur) et ne contiendra que des méthodes de classe (donc `static`). L'une d'elles sera appelée `lireAuteurs`, et contiendra les lignes qui correspondent :
 - a. Au calcul du tableau des auteurs
 - b. A l'insertion de la vue des auteurs

La toute première commande, qui correspond à l'insertion du modèle, sera écrite avant la déclaration de la classe. Réalisez ces opérations.

2. Dans le fichier `routeur.php`, après avoir réalisé l'insertion du fichier contrôleur, insérez la commande qui appelle la méthode `static lireAuteurs()`. Vérifiez que tout fonctionne.

VOUS VENEZ DE METTRE EN PLACE TOUTE LA LOGIQUE MVC.

MAINTENANT, TOUTE ACTION NOUVELLE DEVRA S'ADAPTER A CETTE LOGIQUE !

3. Le contrôleur Acteur va avoir un certain nombre d'actions à sa disposition : `lireAuteurs()` n'est que la toute première. Nous allons donc légèrement modifier le routeur qui va gérer une action récupérée dans l'url, par le biais de `$_GET`, et appeler l'action correspondante du contrôleur. Mettez tout ceci en place. Un appel classique sera :

```
TP3/ex4/routeur.php?action=lireAuteurs
```

4. Mettez en place une valeur par défaut pour l'action, pour éviter que l'appel suivant ne provoque une erreur

```
TP3/ex4/routeur.php
```

La valeur par défaut logique est `$action = lireAuteurs`

5. Que se passe-t-il si on appelle une action non définie ? Essayez par exemple d'appeler le routeur avec l'action `lireTousLesAuteurs...` Pour parer à ce bug, regardez la méthode `get_class_methods` sur le manuel PHP et adaptez le routeur aux méthodes non définies. Vérifiez que tout fonctionne.

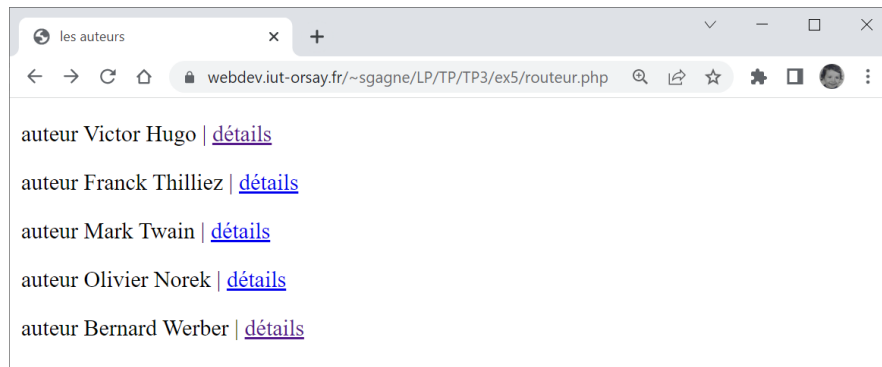
Exercice 5 – l'action lireUnAuteur

Nous allons maintenant produire tout le code qui correspond à l'action `lireUnAuteur` du contrôleur. Commencez par dupliquer votre travail en `TP3/ex5`.

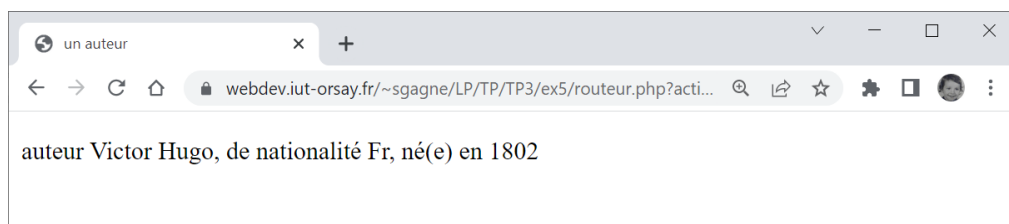
1. Ajoutez cette méthode `static` au contrôleur :

```
public static function lireUnAuteur() {  
    $numAuteur = $_GET["numAuteur"];  
    $auteur = Auteur::getAuteurByNum($numAuteur);  
    require_once("vue/unAuteur.php");  
}
```

2. Cette méthode aura besoin, pour fonctionner, que la méthode `getAuteurByNum` du modèle soit codée, avec bien sûr une requête préparée. Codez cette méthode du modèle.
3. Créez aussi la vue `unAuteur.php` qui n'affiche que les renseignements sur l'auteur en question. Elle ressemble à la vue `lesAuteurs.php`, mais il faut adapter le titre et supprimer la boucle bien sûr.
4. Vous allez maintenant modifier la vue `lesAuteurs.php`. La boucle n'appellera plus la méthode `afficher()`, qu'on réservera à `unAuteur.php`. Maintenant, `lesAuteurs.php` affichera les auteurs, puis un lien détails qui permettra d'aller sur la page `unAuteur.php` avec le bon `numAuteur`. Voici une capture :



5. Modifiez `auteur.php` pour que le descriptif d'un auteur soit complet (hormis son `numAuteur`)



Exercice 6 – un menu plus pratique

Dupliquez tout en TP3/ex6.

Pour pouvoir revenir à la liste des auteurs, vous allez écrire un menu avec un lien (qui pointera vers `routeur.php?action=lireAuteurs`) que vous incluez dans tous les fichiers vue. Vérifiez que tout fonctionne.

Exercice 7 – organisation des vues

Le problème est que l'on va avoir des répétitions de code du menu. Essayez de réfléchir à une solution qui évite ceci.