

Programmation orientée Objet

Class BookManager

En POO l'accès aux données est toujours séparé du reste. On appelle cela un manager.

Nous allons donc :

- Créer une classe BookManager dans un nouveau fichier.
- Elle aura un attribut privé qui sera la liste de nos livres.
- Elle aura une fonction publique pour ajouter des livres qui seront stockés sous forme de tableau dans notre attribut privé.
- Elle aura un getter de l'attribut livres

Une fois fait vous pouvez supprimer l'attribut static dans la classe Livre qui ne sera plus utile et l'enlever du constructeur.

Je vous laisse modifier en conséquence le code de book.php afin d'avoir le même fonctionnement que plus tôt, mais avec notre manager 😊

Création de la bdd

Avoir un manager c'est cool mais temps que les données sont écrits en dures, c'est moins cool...

Créez cette base de données dans phpmyadmin.

Bdd : bibliothèque

Table livres :

- idLivre INT AI PK
- nom varchar 255
- nbPages int
- image varchar 150

Implémentez vos livres en BDD.

Création de la classe Model

Pour préparer l'accès aux données, créez une classe abstract Model :

- Elle aura un attribut private \$pdo. (Nous l'utiliserons pour stocker la connexion à la bdd)
- Une fonction private setDB() qui va stocker la connexion avec PDO dans l'attribut private \$pdo. Exemple :

```
$this->pdo = new PDO("mysql:host=localhost;dbname=bibliotheque-  
poo;charset=utf8","root","");  
$this->pdo->setAttribute(PDO::ATTR_ERRMODE,PDO::ERRMODE_WARNING);
```

- Une fonction protected getDB qui va utiliser setDB si l'attribut \$pdo est null et va retourner cette attribut (\$this->pdo).

Amélioration de BookManager

Pour pouvoir accéder à la liste complète des livres de façon dynamique nous allons ajuster notre classe BookManager comme en y rajoutant ceci :

Une fonction public loadingBooks avec dedans :

- Vous allez créer la requête sql qui permet de récupérer les livres dans la fonction loadingBooks (faites un extends de model sur BookManager afin de pouvoir utiliser la db)
- Il ne vous reste plus qu'à foreach, toujours dans la fonction, le résultat de le requête sql (fetch), de créer des instances de livres à l'aide des infos de la bdd et de réutiliser notre fonction addBook.
- Maintenant vous pouvez supprimez vos livres crée à la main dans books et utiliser en lieu et place notre nouvelle fonction loadingBooks 😊

Pensez à réajuster vos includes par rapport à nos récents changements.