

TP n°4 - Les bases de données

Site: [Université de Reims Champagne-Ardenne](#)

Cours: INFO0303 - Technologies Web 2

Livre: TP n°4 - Les bases de données

Imprimé par: CYRIL RABAT

Date: samedi 23 septembre 2023, 07:44

Table des matières

1. Mise en place d'une base de données
2. La classe MyPDO
3. Connexion d'un utilisateur
4. Forum

1. Mise en place d'une base de données

L'objectif de cet exercice est de mettre en place une base de données qui sera utilisée dans les exercices suivants.

1. Accédez à *phpMyAdmin* (une fois votre serveur Web démarré) et loguez-vous.
 - Par défaut, l'adresse est <http://localhost/phpmyadmin/>
 - Les identifiants par défaut sont « root », sans mot de passe (ou « root / root » suivant les configurations)
2. Créez une nouvelle base de données :
 - Choisissez un nom (tout en minuscules, pas de caractères spéciaux, pas d'espaces) : « tp4 », par exemple ;
 - Pour l'interclassement, choisissez « utf8_general_ci » (par exemple).
3. Créez une nouvelle table nommée « user » contenant 3 champs :
 - « id » de type UNSIGNED INT, cochez « A_I » (pour auto-increment), cela créera automatiquement une clé primaire
 - « login » de type VARCHAR(100)
 - « password » de type VARCHAR(100)
4. Ajoutez manuellement plusieurs utilisateurs dans votre table (onglet « Insérer »).
5. Vérifiez le contenu de la table (onglet « Parcourir »).

Dans une application destinée à la production, les mots de passe doivent être chiffrés. Nous ne nous y intéresserons pas ici. En effet, les *frameworks* PHP (comme *Laravel*) intègrent déjà cette fonctionnalité.

2. La classe MyPDO

Comme vu en TD, nous allons écrire une classe **MyPDO** pour nous aider à créer et maintenir une connexion vers la base de données.

1. Créez un répertoire **config** contenant un fichier nommé **DBConfig.php**.
2. Dans ce fichier de configuration, spécifiez les éléments nécessaires pour se connecter à votre base de données à l'aide de plusieurs constantes :
 - **DB_HOST** : localhost
 - **DB_NAME** : le nom choisi dans l'exercice précédent (tp4 ?)
 - **DB_LOGIN** et **DB_PASSWORD** : les identifiants utilisés pour se connecter à votre base
3. Comme vu en TD, écrivez la classe **MyPDO** (pas dans le répertoire **config**) qui contient l'unique méthode **getInstance** et qui retourne le connecteur PDO vers la base de données.
4. Écrivez un script PHP permettant d'afficher dans une page HTML le contenu de la table **user** :
 - Utilisez la méthode **getInstance** pour récupérer la connexion.
 - Exécutez une requête de type « SELECT » pour sélectionner tous les utilisateurs.
 - Récupérez-les avec la méthode **fetch** et affichez-les.
5. Que se passe-t-il lorsque vous saisissez l'URL « localhost/XXX/config/ » (où « XXX » est le répertoire de votre projet) ?
6. Ajoutez le fichier nommé **.htaccess** dans le répertoire **config** qui possède le contenu suivant :

```
DENY FROM ALL
```

7. Et maintenant, que se passe-t-il si vous essayez d'accéder au répertoire **config** ?

Vous êtes maintenant prêt pour exploiter la base de données dans une application Web.

3. Connexion d'un utilisateur

À l'aide de la base de données que nous avons créée précédemment, nous allons gérer la connexion d'utilisateurs.

Nous allons écrire deux scripts. `login.php` contient le formulaire de connexion et vérifie si les identifiants saisis sont corrects. Si c'est le cas, il redirige l'utilisateur vers le script `index.php`. Ce script ne doit être accessible que si l'utilisateur est connecté (si ce n'est pas le cas, il redirige avec le script de login). Il contient pour le moment un simple message de bienvenue.

1. Rappelez comment faire en sorte que l'utilisateur soit identifié comme étant connecté et comment vérifier si c'est le cas dans le script `index.php`.
2. Écrivez les deux scripts.

4. Forum

Nous souhaitons développer une application de forums de discussion entre utilisateurs. Un forum correspond à un titre et est associé à une catégorie. Il contient également un ensemble de messages. Chaque message est caractérisé par un texte, une date et est associé à l'utilisateur. La catégorie est caractérisée par un simple intitulé.

1. Proposez un MPD correspondant à cette application puis créez les tables et les clés dans votre base de données.
2. Écrivez une classe `Forum` (on ne considère pas les messages).
3. Écrivez la classe CRUD `ForumModel1`. Testez dans un script si toutes les méthodes sont opérationnelles.
4. Faites de même avec la classe `Message`.
5. Liez cette application avec l'exercice précédent : le script `index.php` correspond maintenant à la liste des forums et seuls les utilisateurs connectés y ont accès.