

# TP2

L'objectif de ce TP est de:

- o S'initier aux aspects BD du langage PHP5, au travers de la couche PDO;
- o Expérimenter (et apprécier) la facilité avec laquelle le changement de mode de gestion des données peut être effectué lorsque la partie "Modèle" a été isolée.

## Ajout d'une base de données dans l'application image

### 1. Introduction

Le TP1 vous a permis de mettre en place une application de navigation dans une base d'images dans laquelle la partie de code permettant de gérer les données était isolée. L'intérêt de cette modularité du code est de permettre de modifier le mode de gestion de données facilement car seuls les fichiers du répertoire Modèle seront concernés par le changement. Nous souhaitons remplacer le code existant basé sur le parcours d'un répertoire par une véritable base de données. Cela nous permettra d'ajouter d'autres informations concernant les images.

La base de données que nous allons définir sera dans un premier temps composée d'une seule table « image » où chaque enregistrement décrit une image avec un identifiant numérique, le chemin (path) relatif jusqu'au fichier image, une catégorie et un commentaire :

image (id, path, category, comment)

Exemple de n-uplet :

5	jons/external/53255main_97_jwhires.jpg	NASA	Jupiter
---	--	------	---------

### 2. Installation de la base de données

Afin de créer la base de donnée, récupérez :

- Le fichier **imageDB.txt** qui contient tous les n-uplets sous forme texte
- Le fichier **create.sql** qui contient l'instruction sql permettant de construire la table

Vous pouvez utiliser le gestionnaire de base de données de votre choix dans la mesure où il est compatible avec la couche PDO. Néanmoins, dans un premier temps, nous vous proposons d'utiliser SQLite (qui est intégré à PHP 5). Par la suite, vous pourrez par exemple utiliser MySQL ou PostgreSQL. Grâce à PDO, vous n'aurez qu'à changer une ligne dans le code de l'application pour changer de SGBD.

Une base de donnée de SQLite est un simple fichier et n'est associée à aucun serveur. Téléchargez (<http://www.sqlite.org/download.html>) sqlite3, l'interpréteur de commandes SQLite (<http://www.sqlite.org/cli.html>). Choisissez l'exécutable correspondant à votre environnement et placez le dans le répertoire contenant imageDB.txt et create.sql. Lancez l'interpréteur et tapez les commandes suivantes :

1. Pour créer la table image :

```
sqlite> .read create.sql
```

2. Pour le chargement des données il faut spécifier le caractère de séparation des champs (la tabulation), puis réaliser l'opération de chargement :

```
sqlite> .separator "\t"
```

```
sqlite> .import imageDB.txt image
```

3. Pour sauver ensuite la base créée (qui s'appelle "main" par défaut) dans un fichier (par exemple "imageDB"):

```
sqlite> .backup main imageDB
```

Et déposez ce fichier (imageDB) à l'endroit souhaité de votre application Web image.

Pour obtenir l'aide de l'interpréteur : `sqlite>.help`

### 3.Modification du modèle

Vous allez faire évoluer le code de votre application. Nous vous conseillons de conserver la version précédente et de faire évoluer une copie de votre code dans un autre répertoire.

Nous vous proposons d'utiliser la couche générique PDO qui unifie l'accès à plusieurs types de base de données. Vérifiez que l'accès PDO est bien installé en examinant le résultat de `phpinfo()`.

#### PDO

PDO support	enabled
PDO drivers	mysql, sqlite

Seules 3 méthodes de la classe ImageDAO nécessitent d'être modifiées. Il s'agit de **\_\_construct**, **getImage** et **size**. Pensez aussi à ajouter à la classe ImageDAO, l'attribut privé `$db` qui contiendra la connection à la base de données. Avec SQLite, il faut indiquer le chemin absolu du fichier dans la description du "data source name", c'est à dire le chemin sur le fichier de la base de donnée (créé précédemment, voir ci-dessus). Notez qu'aucune identification et mot de passe ne sont nécessaires avec SQLite.

Le constructeur de la classe ImageDAO correspondra à la connexion à la base de données :

```
function __construct() {
    $dsn = 'sqlite:c:\...\wamp\www\...\imageDB'; // Data source name
    $user= ''; // Utilisateur
    $pass= ''; // Mot de passe
    try {$this->db = new PDO($dsn, $user, $pass); //$db est un attribut privé d'ImageDAO
    } catch (PDOException $e) { die ("Erreur : ".$e->getMessage());
    }
}
```

La recherche d'une image à partir de son id se fait par un appel à la base de données :

```
function getImage($id) {
    $s = $this->db->query('SELECT * FROM image WHERE id='.$id);
    if ($s) {
        à compléter ...
    } else {
        print "Error in getImage. id=".$id."<br/>";
        $err= $this->db->errorInfo();
        print $err[2]."<br/>";
    }
}
```

Le calcul du nombre d'images se fait par un appel à la base de données :

```
function size() { {
    à compléter;
    indication : on pourra se servir de la fonction count qui retourne le nombre d'éléments d'un
    tableau }
```

On pourra vérifier que l'objet imageDAO a bien été initialisé en utilisant le test unitaire via l'url:

<http://localhost/.../image/imageDAO.php?test>

Enlevez tout ce qui concerne l'ancienne structure d'accès aux images et vérifiez que votre application fonctionne toujours correctement.

### 4.Utilisation des catégories et des commentaires

Vous avez remarqué que la base de données des images contient deux nouveaux champs : une catégorie, groupant plusieurs images, et un commentaire par image.

Complétez votre code pour faire apparaître ces deux informations à l'affichage des images. Cette fois-ci, vous devez donc compléter le modèle et le code correspondant à l'affichage des informations.



*Faut-il une mémoire d'éléphant pour faire du PHP ?  
Non, pas avec un WEB disponible ! Bon travail ...*