

Travaux dirigés n° 3

PHP et les bases de données

Exercice 1 (Utilisation des bases de données)

Nous souhaitons écrire une classe permettant de gérer les connexions à la base de données d'un site. Les identifiants (l'URL du SGBD, le nom de la base de données, le login et le mot de passe utilisateur) sont spécifiés dans un fichier PHP `DBConfig.php`.

1. Donnez le contenu du fichier `DBConfig.php`.
2. Rappelez les précautions à prendre concernant ce fichier.
3. Écrivez la classe `MyPDO` contenant uniquement la méthode statique `getInstance` qui permet de créer ou retourner l'instance de l'objet PDO permettant d'accéder à la base de données.
4. Expliquez comment utiliser cette classe et donnez un exemple d'utilisation en affichant un enregistrement spécifique et le contenu complet d'une table quelconque.

Exercice 2 (Actualités)

Nous reprenons les actualités vues dans un TD précédent. Elles sont maintenant stockées dans une base de données.

1. Expliquez quelles sont les tables nécessaires pour cette application et spécifiez les champs avec leur types. Indiquez les clés primaire et secondaire(s).
 2. Donnez le contenu de la classe `Actualite` qui nous permettra de représenter un enregistrement.
 3. Écrivez la classe `ActualiteModel` correspondant à la classe CRUD (pour `Create`, `Read`, `Update` et `Delete`) de la classe `Actualite`. Elle contient également une méthode `fromArray` permettant de créer un objet `Actualite` à partir d'un tableau associatif.
- N.B. : vous utilisez la classe `MyPDO` de l'exercice précédent.

Exercice 3 (Actualités dynamiques)

Nous souhaitons réaliser un site où des actualités sont affichées dynamiquement une par une (avec un changement toutes les 3 secondes, par exemple). Pour cela, nous utilisons AJAX et les éléments développés dans l'exercice précédent.

1. Donnez tous les éléments à mettre en place.
2. Comment mémoriser l'actualité courante ?
3. Donnez l'ordre d'appel et toutes les requêtes échangées sous la forme d'un diagramme d'échange.
4. Donnez le contenu du script PHP et de la partie *Javascript*.
5. Pensez-vous qu'AJAX soit une bonne solution pour cette application ? Expliquez.

Annexes - Manuel PHP

Classe **PDO** (version simplifiée) :

```
public __construct(string $dsn [, string $username [, string $passwd  
                    [, array $options ]]])  
public int exec(string $statement)  
public string lastInsertId([ string $name = NULL ])  
public PDOStatement prepare(string $statement  
                             [, array $driver_options = array() ])  
public PDOStatement query(string $statement)
```

Représente une connexion entre PHP et un serveur de base de données (SGBD). À noter que le paramètre `$dsn` du constructeur correspond à la chaîne suivante (pour une connexion à un SGBD *mySQL*) :

```
mysql:host=localhost;dbname=mabase;charset=utf8
```

Classiquement, les options utilisées (paramètre `$driver_options`) sont les suivantes :

```
$driverOptions = [  
    PDO::ATTR_ERRMODE          => PDO::ERRMODE_EXCEPTION,  
    PDO::ATTR_DEFAULT_FETCH_MODE => PDO::FETCH_ASSOC  
];
```

Classe **PDOStatement** (version simplifiée) :

```
public bool bindColumn(mixed $column, mixed &$param [, int $type  
                    [, int $maxlen [, mixed $driverdata ]]])  
public bool bindParam(mixed $parameter, mixed &$variable  
                    [, int $data_type = PDO::PARAM_STR [, int $length  
                    [, mixed $driver_options ]]])  
public bool bindValue(mixed $parameter, mixed $value  
                    [, int $data_type = PDO::PARAM_STR ])  
public bool closeCursor(void)  
public int columnCount(void)  
public void debugDumpParams(void)  
public bool execute([ array $input_parameters ])  
public mixed fetch([ int $fetch_style  
                    [, int $cursor_orientation = PDO::FETCH_ORI_NEXT  
                    [, int $cursor_offset = 0 ]]])  
public array fetchAll([ int $fetch_style [, mixed $fetch_argument  
                    [, array $ctor_args = array() ]]])  
public mixed fetchColumn([ int $column_number = 0 ])  
public mixed fetchObject([ string $class_name = "stdClass" [, array $ctor_args ]])  
public mixed getAttribute(int $attribute)  
public array getColumnMeta(int $column)  
public bool nextRowset(void)  
public int rowCount(void)  
public bool setAttribute(int $attribute, mixed $value)  
public bool setFetchMode(int $mode)
```

Représente une requête préparée et, une fois exécutée, le jeu de résultats associé.