



## Préambule

Ce document a pour but de faire un rappel sur l'objet *PHP PDO* (*PHP Data Objects*) qui permet de se connecter facilement à des bases de données afin d'y effectuer des requêtes. On y trouve :

- La connexion à la base de données.
- Les requêtes dites préparées.
- La déconnexion de la base de données.

## 1 Connexion à la base de données

```
try
{
    $db = new PDO('mysql:host=SERVER_NAME;dbname=DATABASE;charset=utf8',
        LOGIN, PASSWORD);
    $db->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
}
catch (PDOException $exception)
{
    error_log('Connection error: '.$exception->getMessage());
}
```

Où les éléments en majuscules doivent être respectivement remplacés par :

- L'*ip* ou le nom de la machine où se trouve la base de données.
- Le nom de la base de données.
- Le *login* de connexion à la base de données.
- Le mot de passe de connexion à la base de données.

Dans le cas où ces informations sont correctes, la variable `$db` contient l'identifiant permettant de manipuler la base de données.

## 2 Requêtes préparées sans paramètre

```
try
{
    $request = 'SELECT * FROM users';
    $statement = $db->prepare($request);
    $statement->execute();
    $result = $statement->fetchAll(PDO::FETCH_ASSOC);
}
catch (PDOException $exception)
{
    error_log('Request error: '.$exception->getMessage());
}
echo($result['login']);
```

Les requêtes dites « préparée », permettent d'interroger une base de données en toute sécurité. Pour cela il est nécessaire de suivre les étapes suivantes :

- Création de la requête.
- Préparation de la requête.
- Exécution de la requête.
- Récupération des résultats.

Notez que pour récupérer la valeur d'un champ particulier, on utilise la notion de tableau associatif où la clé est le nom du champ voulu.

### 3 Requêtes préparées avec paramètres

```
$name = 'jean';
$age = 21;

try
{
    $request = 'SELECT * FROM users WHERE name=:name and age=:age';
    $statement = $db->prepare($request);
    $statement->bindParam(':name', $name, PDO::PARAM_STR, 20);
    $statement->bindParam(':age', $age, PDO::PARAM_INT);
    $statement->execute();
    $result = $statement->fetchAll(PDO::FETCH_ASSOC);
}
catch (PDOException $exception)
{
    error_log('Request error: '.$exception->getMessage());
}
echo($result['login']);
```

Parfois il est nécessaire d'associer des paramètres, ou variables, à notre requête. Cependant, le risque est que ces variables contiennent du code *SQL* qui risque d'engendrer une « injection *SQL* ». Afin d'éviter cela, *PDO* propose d'utiliser des *placeholder* (ici `:name` et `:age`) qui seront ensuite remplacés par la valeur des variables grâce à la fonction `bindParam`. Noter que le nom de ces *placeholder* est libre. Dans la fonction `bindParam` on spécifie le nom du *placeholder* ainsi que le nom de la variable de remplacement. Il est aussi possible de spécifier le type de la variable ainsi que sa taille s'il s'agit d'une chaîne de caractères.

Notez que `fetch(PDO::FETCH_ASSOC)` permet de renvoyer le premier résultat alors que `fetchAll(PDO::FETCH_ASSOC)` permet de retourner l'ensemble des résultats.

### 4 Déconnexion de la base de données

```
unset($db);
```