



Programmation orientée objet en PHP

[Accueil](#) ▶ [Mes cours](#) ▶ [Développement logiciel](#) ▶ [POO PHP](#) ▶ Les bases de données avec PDO ▶ [PDO : PHP Data Object](#)

PDO : PHP Data Object

PDO est une couche d'abstraction qui permet de travailler avec (presque) n'importe quel type de Service de Gestion de Base de Données. PDO définit un certain nombre de classes qui permettent d'effectuer des traitements de persistance en se reposant sur des drivers (ou pilotes) spécifiques à chaque fournisseur de SGBD. L'extension PDO et les drivers doivent être activés si l'on souhaite utiliser PDO.

Le travail avec les SGBD se fait de façon transparente et homogène : quelque soit le SGBD, on utilise les mêmes objets et méthodes fournis par PDO.

L'extension PDO propose trois classes principales :

- **PDO** : qui réalisera le travail de persistance
- **PDOStatement** : qui encapsulera les requêtes et leurs résultats
- **PDOException** : exception déclenchée par les méthodes de PDO et PDOStatement.

La connexion à la base de donnée se fait lors de l'instanciation d'un objet PDO. Le constructeur de PDO prendra en paramètre trois chaînes de caractères :

- la connexion ou data source : cette chaîne contiendra l'hôte et le nom du schéma
- l'utilisateur de la base
- le mot de passe de l'utilisateur.

Le constructeur pourra aussi prendre en paramètre un tableau contenant diverses options de connexion. En cas de problème de connexion à la base une PDOException est levée.

Nous allons nous connecter à une base de donnée MySQL : le schéma est "test_pdo", le nom de l'utilisateur est "root" et le mot de passe est "root" (l'utilisateur et le mot de passe sont peut-être différents pour vous).

```
1 <?php
2
3 $dsn = 'mysql:host=localhost;dbname=test_pdo';
4 $username = 'root';
5 $passwd = 'root';
6 try {
7     $conn = new PDO($dsn, $username, $passwd,
8         [PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION]);
9 } catch (PDOException $ex) {
10     echo $ex->getMessage();
11 }
```

Nous avons ajouté au constructeur un tableau avec une option de configuration. Les options sont indiquées sous forme de couples clé/valeur. La clé et la valeur utilisées ici sont des constantes de la classe PDO. Cette option indique que les erreurs rencontrées par les classes PDO devront être traitées sous formes d'exceptions.

La classe PDO possède deux méthodes pour effectuer des requêtes simples, ces deux méthodes prennent en paramètre une requête SQL sous forme de chaîne de caractères :

- **exec()** : cette méthode exécute une requête et renvoie le nombre d'enregistrements affectés
- **query()** : cette méthode exécute une requête et renvoie un objet de type PDOStatement qui encapsule le résultat de la requête

La classe PDOStatement possède deux méthodes pour récupérer les résultats de la requête :

- **fetch()** : renvoie les résultats ligne à ligne
- **fetchAll()** : renvoie tous les résultats dans un tableau.

Les méthodes **fetch()** et **fetchAll()** peuvent prendre en paramètre un **fetch_style** sous forme de constante qui détermine la façon dont PDO retourne les résultats :

- PDO::FETCH_ASSOC → retourne un tableau associatif indexé par le nom de la colonne, comme retourné dans le jeu de résultats
- PDO::FETCH_BOTH → retourne un tableau indexé par les noms de colonnes mais aussi par les numéros de colonnes (commençant à l'indice 0), comme retournés dans le jeu de résultats
- PDO::FETCH_OBJ → retourne un objet anonyme avec les noms de propriétés qui correspondent aux noms des colonnes retournées dans le jeu de résultats

Par défaut le fetch est de type PDO::FETCH_BOTH.

La classe PDOStatement possède aussi une méthode closeCursor() qui permet de libérer la connexion du serveur, permettant ainsi à d'autres requêtes SQL d'être exécutées,

Voici un exemple ce script PHP contenant du HTML et affichant les personnes extraites de la base de données :

```

1  <?php
2  // index.php
3  $dsn = 'mysql:host=localhost;dbname=test_pdo';
4  $username = 'root';
5  $passwd = 'root';
6  try {
7      $conn = new PDO($dsn, $username, $passwd,
8          [PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION]);
9      $query = 'SELECT * FROM personne';
10     $stmt = $conn->query($query);
11     $result = $stmt->fetchAll(PDO::FETCH_ASSOC);
12 } catch (PDOException $ex) {
13     header('Location: error_page.html');
14 }
15 ?>
16 <!DOCTYPE html>
17 <html>
18     <head>
19         <meta charset="UTF-8">
20         <title>Les personnes</title>
21     </head>
22     <body>
23         <table>
24             <tr>
25                 <th>Id</th>
26                 <th>Nom</th>
27                 <th>Prénom</th>
28             </tr>
29             <?php foreach ($result as $personne): ?>
30                 <tr>
31                     <td><?php echo $personne['id'] ?></td>
32                     <td><?php echo $personne['nom'] ?></td>
33                     <td><?php echo $personne['prenom'] ?></td>
34                 </tr>
35             <?php endforeach; ?>
36         </table>
37     </body>
38 </html>

```

En cas d'exception, le script redirige vers une page d'erreur avec un code erreur HTTP (500).

errorPage.html

```

1  <!DOCTYPE html>
2  <html>
3      <head>
4          <title>Erreur</title>
5          <meta charset="UTF-8">
6          <meta name="viewport" content="width=device-width">
7      </head>
8      <body>
9          <div>Oups...</div>
10     </body>
11 </html>

```

Voici un autre exemple de script PHP affichant les personnes extraites de la base de données :

```

1  <?php
2  // index.php
3  $dsn = 'mysql:host=localhost;dbname=test_pdo';
4  $username = 'root';
5  $passwd = 'root';
6  try {
7      $conn = new PDO($dsn, $username, $passwd,
8          [PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION]);
9      $query = 'SELECT * FROM personne';
10     $stmt = $conn->query($query);
11 } catch (PDOException $ex) {
12     header('Location: error_page.html');
13 }
14 ?>
15 <!DOCTYPE html>
16 <html>
17     <head>
18         <meta charset="UTF-8">
19         <title>Les personnes</title>
20     </head>
21     <body>
22         <table>
23             <tr>
24                 <th>Id</th>
25                 <th>Nom</th>
26                 <th>Prénom</th>
27             </tr>
28             <?php while ($personne = $stmt->fetch(PDO::FETCH_ASSOC)) : ?>
29                 <tr>
30                     <td><?php echo $personne['id'] ?></td>
31                     <td><?php echo $personne['nom'] ?></td>
32                     <td><?php echo $personne['prenom'] ?></td>
33                 </tr>
34             <?php endwhile; ?>
35         </table>
36     </body>
37 </html>

```

Nous allons maintenant ajouter une personne à notre base de donnée. Nous créons donc un formulaire de saisie et un script d'ajout à la base.

formulaire.html

```

1  <!DOCTYPE html>
2  <html>
3      <head>
4          <title>Ajout d'une personne</title>
5          <meta charset="UTF-8">
6          <meta name="viewport" content="width=device-width">
7      </head>
8      <body>
9          <form action="add_personne.php" method="post">
10              <label>Nom : <input type="text" id="nom" name="nom"></label><br>
11              <label>Prénom :
12                  <input type="text" id="prenom" name="prenom">
13              </label><br>
14              <input type="submit" value="Ajouter">
15          </form>
16      </body>
17 </html>

```

```

1  <?php

```

```

2
3 $dsn = 'mysql:host=localhost;dbname=test_pdo';
4 $username = 'root';
5 $passwd = 'root';
6 try {
7     $conn = new PDO($dsn, $username, $passwd,
8         [PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION]);
9     $nom = $_POST['nom'];
10    $prenom = $_POST['prenom'];
11    $query = "INSERT INTO personne (nom, prenom) VALUES ('$nom', '$prenom')";
12    $conn->exec($query);
13    header('Location: index.php');
14 } catch (PDOException $ex) {
15     header('Location: error_page.html');
16 }

```

Fin

NAVIGATION



Accueil

■ Ma page

Pages du site

Mon profil

Cours actuel

POO PHP

Participants

Généralités

La programmation orientée objet : premiers pas

L'héritage

Les interfaces

Le typage

Les namespaces

Les exceptions

Les bases de données avec PDO

PDO : PHP Data Object

personne.sql

La classe MySQLDAO

Les requêtes préparées

T.P. MySQLDao

Les tests avec PHPUnit

Petite application version 2

Petite application version 3

[Mes cours](#)

ADMINISTRATION



Administration du cours

Réglages de mon profil

Connecté sous le nom « [Arnaud Lemais](#) » ([Déconnexion](#))
[POO PHP](#)