

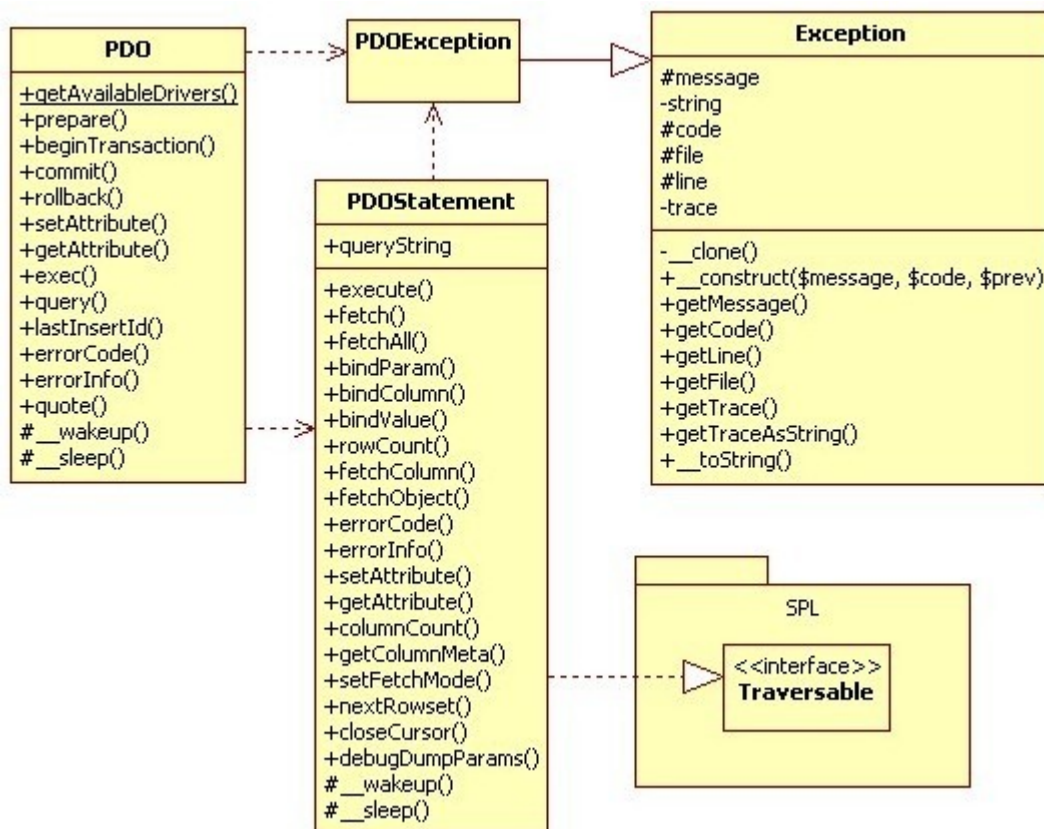
# PDO (PHP Data Objects)

## Interface d'accès aux Bases de Données

PDO fournit une interface d'abstraction à l'accès de données, ce qui signifie que vous utilisez les mêmes fonctions pour exécuter des requêtes ou récupérer les données quelque soit la base de données utilisée.

Cette extension PHP est incluse dans la distribution standard depuis PHP 5.1.

Elle donne pour cela accès à 3 classes :



1. **PDO** : la classe d'interaction principale avec le SGBD ;
2. **PDOStatement** : la classe gérant les résultats de requêtes, préparées, ou exécutées ;
3. **PDOException** : classe d'exception.

### Activer PDO

#### Avec Wamp

Normalement, PDO est activé par défaut. Pour le vérifier, faites un clic gauche sur l'icône de WAMP dans la barre des tâches, puis allez dans le menu PHP / Extensions PHP et vérifiez que `php_pdo_mysql` est bien coché.

#### Avec EasyPHP

```

1 ;extension=php_pdo_firebird.dll
2 ;extension=php_pdo_mssql.dll
3 extension=php_pdo_mysql.dll
4 ;extension=php_pdo_oci.dll
5 ;extension=php_pdo_odbc.dll
    
```

Se connecter à MySQL avec PDO

```
$bdd = new PDO('mysql:host=localhost;dbname=BaseClients', 'root', '');
```

## Ou

Placer les paramètres de connexion dans un fichier séparé : connect.php

```
<?php  
  
$host= "localhost";  
  
$util= "root";  
  
$password= "";  
  
$bdd= "BaseClients";  
  
?>
```

Dans votre script PHP

```
require("connect.php"); // on peut également utiliser include  
  
$pdo_options[PDO::ATTR_ERRMODE] = PDO::ERRMODE_EXCEPTION;  
  
$bdd = new PDO('mysql:host='.$host.';dbname='.$bdd, $util, $password,  
$pdo_options)
```

## Les erreurs

```
<?php  
require("connect.php");  
try  
{  
    $pdo_options[PDO::ATTR_ERRMODE] = PDO::ERRMODE_EXCEPTION;  
    $bdd = new PDO('mysql:host='.$host.';dbname='.$bdd, $util, $password,  
$pdo_options);  
}  
catch (Exception $e)  
{  
    die('Erreur : ' . $e->getMessage());  
}  
?>
```

PHP exécute les instructions à l'intérieur du bloc try{ } . Si une erreur se produit, il s'arrête et exécute la partie se situant dans le catch{ }.

Le tableau \$pdo\_options permet d'activer les exceptions PDO. On peut ainsi récupérer les erreurs lorsqu'elles se produisent.

Exemple de parcours d'une table CLIENT dont le schéma relationnel est le suivant :

CLIENTS(IdClient, NomClient, PrenomClient, VilleClient)  
IdClient : clé primaire

```
<?php
require("connect.php");
try
{
?><table>
<tr><td>Numéro Client</td><td>Nom Client</td><td>Prénom Client</td><td>Ville Client</td></tr>
<?php
    $pdo_options[PDO::ATTR_ERRMODE] = PDO::ERRMODE_EXCEPTION;
    $bdd = new PDO('mysql:host='.$host.';dbname='.$bdd, $util, $password, $pdo_options);
    $reponse = $bdd->query('select * from Clients'); //Exécute la requête
    while ($donnees = $reponse->fetch()) //parcours le curseur pour récupérer les données
    {
?>
        <tr><td><?php echo $donnees["IdClient"]?></td><td><?php echo $donnees["NomClient"]?>
        <td><?php echo $donnees["PrenomClient"] ?></td><td><?php echo $donnees["VilleClient"]?>
    </td></tr>
    <?php
    }
    ?>
    </table>
    <?php $reponse->closeCursor(); // ferme le curseur
}
catch (Exception $e)
{
    die('Erreur : ' . $e->getMessage());
}
?>
```

## **Les requêtes préparées**

Si Vous souhaitez obtenir les clients habitant Montpellier, vous risquez d'avoir à écrire :

```
$reponse = $bdd->query('SELECT NomClient, PrenomClient FROM CLIENTS
WHERE VilleClient=\' . $_POST['ville'] . \'');
```

Vous pouvez préparer votre requête et l'exécuter de cette manière :

```
$req = $bdd->prepare('SELECT NomClient, PrenomClient FROM CLIENTS
WHERE VilleClient = ?');

$req->execute(array($_POST['ville']));
```

Pour obtenir les clients de nom Dupont et habitant Montpellier :

```
<?php  
  
$req = $bdd->prepare('SELECT NomClient, PrenomClient FROM CLIENTS  
WHERE NomClient = ? AND VilleClient = ?');  
  
$req->execute(array($_POST['nom'], $_POST['ville']));  
  
?>
```

Si la requête contient plusieurs variables, il est possible de nommer les marqueurs plutôt que d'utiliser des points d'interrogation.

```
<?php  
  
$req = $bdd->prepare('SELECT NomClient, PrenomClient FROM CLIENTS  
WHERE NomClient = :nom AND VilleClient = :ville');  
  
$req->execute(array('nom' => $_POST['nom'], 'ville' => $_POST['ville']));  
  
?>
```

*Avantage des requêtes préparées : évite les injections SQL.*

Documentation officielle : <http://fr.php.net/pdo>

# TP PHP PDO

## **Exercice 1 :**

A partir de la base de données BaseProduits dont le schéma relationnel est le suivant :

CATEGORIE(CodeCateg, NomCateg)

CodeCateg : clé primaire

PRODUIT(CodeProduit, NomProduit, CategProduit)

CodeProduit : clé primaire

CategProduit : clé étrangère en référence à CodeCateg de la table

CATEGORIE

Réaliser une fonction php nommée AfficheProduits() permettant d'afficher la liste des produits (Code et Nom) d'une catégorie dont le code est passé en paramètre. Si le code passé en paramètre est 0, on affichera la liste complète des produits.

## **Exercice 2**

A partir de la base de données BaseEtudiants dont le schéma relationnel est le suivant :

CLASSE(CodeClasse, LibelleClasse)

CodeClasse : Clé primaire

ETUDIANT(NumEtu, NomEtu, DateNaissanceEtu, ClasseEtu)

NumEtu : clé primaire

ClasseEtu : clé étrangère en référence à CodeClasse de la table CLASSE

Créer la table sous MySQL et insérer des données constituant votre jeu d'essai

Ecrire et tester les fonctions suivantes :

GetEtudiants() qui affiche tous les étudiants

GetEtudiantsClasse() qui affiche les étudiants d'une classe (Le code classe est passé en paramètre)

Les affichages se feront de façon tabulaire.

## **Exercice 3**

Ecrire et tester les trois fonctions suivantes :

RemplitListe() qui remplit une liste déroulante avec les libellés des classes (la valeur correspondant au code de la classe)

CaseACocher() qui affiche les classes sous forme de cases à cocher

BoutonRadio() qui affiche les classes sous forme de boutons radio.

## **Exercice 4**

Ecrire et tester les trois fonctions suivantes :

InsertClasse() qui permet d'insérer une nouvelle classe dans la table CLASSE .

Le code et le libellé de la classe sont passés en paramètre.

ModifClasse() qui modifie le libellé d'une classe. Le code de la classe et le nouveau libellé sont passés en paramètre.

SupprimeClasse()qui supprime une classe. Le code de la classe est passé en paramètre.