

PHP – Les bases de données

1

1

Qu'est-ce que PDO ?

- PDO (PHP Data Objects) permet d'utiliser une base de données en programmant avec un style orienté objet, et qui permet de s'affranchir du SGBD
- PDO est une extension PHP
- Il faut installer le pilote PDO générique ainsi que chacun des pilotes de SGBD que l'on souhaite utiliser

Comment savoir quels pilotes sont chargés pour PDO sur mon hébergement ?

```
<?php
    foreach(get_loaded_extensions() as
$strExtension) {

        if(strpos(strtolower($strExtension),
'pdo') !== FALSE) {

            echo
$strExtension.'<br/>';

        }

    }

?>
```

2

2

Se connecter

```
try{  
    // Connexion à la base de données  
    $db = new PDO("mysql:host=localhost;dbname=nombdd", "user", "pwd");  
  
    // Configuration des exceptions  
    $db->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);  
} catch(Exception $e) {  
    echo "Échec : " . $e->getMessage();  
}
```

3

3

Quelques fonctions PDO

- PDO::exec — Exécute une requête SQL et retourne le nombre de lignes affectées
- PDO::lastInsertId — Retourne l'identifiant de la dernière ligne insérée ou la valeur d'une séquence
- PDO::prepare — Prépare une requête à l'exécution et retourne un objet
- PDO::query — Exécute une requête SQL, retourne un jeu de résultats en tant qu'objet PDOStatement

4

4

Exemple de requête

```
// préparation de la requête
$select = $db->prepare("SELECT id, name, password FROM user");
// Définit le mode de récupération par défaut pour cette requête
$select->setFetchMode(PDO::FETCH_ASSOC);
// Exécution de la requête
$select->execute();

// Récupération de tous les éléments de la requête
$users = $select->fetchAll();

// Affichage des résultats
print_r($users);
```

5

5

Les fetch modes de PDO

- **FETCH_BOTH**
 - mode de récupération par défaut de PDO
 - contient les résultats en double : une fois indexés par le nom des colonnes et une deuxième fois indexés par leur numéro
- **FETCH_NUM**
 - Le tableau retourné contient les résultats indexés par le numéro de la colonne.
- **FETCH_ASSOC**
 - Le tableau retourné contient les résultats indexés par le nom de la colonne

6

6

Requête non préparée

- `query()` permet de retourner un jeu de résultats (SELECT)
`$strQuery = "SELECT * FROM user WHERE id=1;";`
`$arrResult = $db->query($strQuery)->fetch ();`
- `exec()` retourne un nombre de lignes affectées (INSERT, UPDATE, ...)
`$strQuery = "DELETE FROM user WHERE id=1;";`
`$intRowCount = $db->exec($strQuery);`

7

7

Requête préparée

- Une requête préparée est un moule, permettant d'exécuter plusieurs fois la requête avec des paramètres différents

//Préparer la requête

```
$query = "SELECT * FROM foo  
WHERE id=? AND cat=? LIMIT ?;";
```

```
$prep = $pdo->prepare($query);
```

//Associer des valeurs aux place holders

```
$prep->bindValue(1, 120, PDO::PARAM_INT);  
$prep->bindValue(2, 'bar', PDO::PARAM_STR);  
$prep->bindValue(3, 10, PDO::PARAM_INT);
```

//Compiler et exécuter la requête

```
$prep->execute();
```

//Récupérer toutes les données retournées

```
$arrAll = $prep->fetchAll();
```

//Clôre la requête préparée

```
$prep->closeCursor();
```

```
$prep = NULL;
```

8

8

Réutiliser une requête préparée

```
$query = "INSERT INTO foo (nom, prix) VALUES (?, ?);";
$prep = $pdo->prepare($query);

$prep->bindValue(1, "item 1", PDO::PARAM_STR);
$prep->bindValue(2, 12.99, PDO::PARAM_FLOAT);
$prep->execute();

$prep->bindValue(1, "item 2", PDO::PARAM_STR);
$prep->bindValue(2, 7.99, PDO::PARAM_FLOAT);
$prep->execute();

$prep->bindValue(1, "item 3", PDO::PARAM_STR);
$prep->bindValue(2, 17.94, PDO::PARAM_FLOAT);
$prep->execute();

$prep->closeCursor();
$prep = NULL;
```

9

9

Les place holders

- PDO permet l'utilisation de place holders nommés au lieu d'utiliser des points d'interrogation et un indicateur de position numérique

```
$query = 'SELECT * FROM foo WHERE id=:id AND cat=:categorie LIMIT :limit;';
$prep = $pdo->prepare($query);

$prep->bindValue(':limit', 10, PDO::PARAM_INT);
$prep->bindValue(':id', 120, PDO::PARAM_INT);
$prep->bindValue(':categorie', 'bar', PDO::PARAM_STR);

$prep->execute();
$arrAll = $prep->fetchAll();
```

10

10

nombre de lignes retournées

```
$strQuery = 'SELECT * FROM foo;';  
$prep = $pdo->prepare($strQuery);  
$prep->execute();  
  
echo $prep->rowCount() . ' résultat(s)';
```

11

11