

R3.01

PDO

PDO

PHP Data Objects (PDO) est une extension définissant l'interface pour accéder à une base de données avec [PHP](#) (wikipedia)

Une couche d'abstraction sur le SGBD ou le moteur

PDO

Classes :

- PDO la connexion à la BD
- PDOStatement la requête et son résultat
- PDOException une exception

PDO : connexion

Dsn : Data Source Name

```
$connexion = new  
PDO("mysql:host=mon_serveur;dbname=ma_bdd;port=mon_port",  
"mon_identifiant", "mon_mdp");
```

```
$connexion = new PDO("sqlite:/opt/databases/mydb.sq3", NULL,  
NULL);
```

```
$connexion = new PDO(":memory:", NULL, NULL);
```

PDO : Singleton (Cf CM2)

```
class SPDO
{
    private $connexion;
    private static $PDOInstance;

    private function __construct(string $dsn,
                                   ?string $username = null,
                                   ?string $password = null,
                                   ?array $options = null,
                                   ?string $exec = null)
    {
        $this->connexion = new \PDO($dsn,$username , $password, $options);
        $this->connexion->exec($exec);
    }
}
```

PDO : Singleton

```
public static function getInstance(string $dsn,  
                                ?string $username = null,  
                                ?string $password = null,  
                                ?array $options = null, //option spécifiques à un SGBD  
                                ?string $exec = null):SPDO //exécution après connexion  
{  
    if(is_null(self::$PDOInstance))  
    {  
        self::$PDOInstance = new SPDO($dsn,$username , $password, $options, $exec);  
    }  
    return self::$PDOInstance;  
}  
  
public function getConnexion(): PDO{  
    return $this->connexion;  
}  
}
```

PDO : connexion avec singleton

```
const CFG = array(
    "db" => array(
        "host" => HOME."data".DIRECTORY_SEPARATOR,
        "port" => null,
        "database" => "madb.db",
        "login" => "",
        "password" => "",
        "options" => array(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION),
        "exec" => "PRAGMA foreign_keys = ON;"
    )
);
```

```
$connexion = SPDO::getInstance($dsn,CFG["db"]["login"],CFG["db"]["password"],CFG["db"]
["options"],CFG["db"]["exec"]) ->getConnexion();
```

Les requêtes

- Non préparées
 - À éviter car risque d'injection SQL
- Préparées
 - prepare avec paramètres
 - bindparam (par référence) ou bindvalue (par valeur)
 - exec
 - puis parcours (fetch, fetch All, fetchObject, ...)

Exemple

- User avec SQLite

```
CREATE TABLE IF NOT EXISTS user (  
    login TEXT NOT NULL PRIMARY KEY,  
    password TEXT NOT NULL  
);
```

```
INSERT OR REPLACE INTO user (login,password) VALUES ("jojo","pass");  
INSERT OR REPLACE INTO user (login,password) VALUES ("raoul","pass");
```

User avec sqlite

```
$pdo = new  
PDO("sqlite:./madb.db");  
$stmt = $pdo->prepare("SELECT *  
from user");  
$stmt->execute();  
echo "<pre>";  
var_dump($stmt->fetchAll());  
echo "</pre>";
```

Données au format indexé et
associatif

```
array(1) { [0]=>  
array(4) {  
    ["login"]=> string(5) "raoul"  
    [0]=> string(5) "raoul"  
    ["password"]=> string(4) "pass"  
    [1]=> string(4) "pass" } }
```

User avec sqlite

```
$stmt-  
>fetchAll(PDO::FETCH_ASSOC)
```

Données au format associatif

```
array(1) { [0]=>  
  array(2) {  
    ["login"]=> string(5) "raoul"  
    ["password"]=> string(4) "pass"  
  }  
}
```

User avec sqlite

```
class User {}
```

```
$stmt->fetchAll(PDO::FETCH_CLASS,"User")
```

La classe est vide, elle est peuplée avec `__set`, méthode magique invoquée lors de l'accès à une propriété non existante ou inaccessible

Données au format associatif

```
array(1) { [0]=> object(User)#3 (2) { ["login"]=> string(5) "raoul" ["password"]=> string(4) "pass" } }
```

User avec sqlite

```
class User {  
    private string $login;  
    private string $password;  
}
```

// Toujours OK

// Si ajout d'un constructeur pb

Données au format objet

```
array(1) { [0]=> object(User)#3 (2)  
  { ["login"]=> string(5) "raoul"  
    ["password"]=> string(4) "pass" } }
```

Ajout

```
$stmt = $pdo->prepare("insert  
into user (login, password) values  
( 'Roméo', 'pass' );");  
$stmt->execute();
```

```
select * from user ;  
jojo | pass  
raoul | pass  
Roméo | pass
```

Ajout

```
$login="rachida";  
$pass="pass";  
$stmt = $pdo->prepare("insert into user (login, password) values  
(:login,:pass)");  
$stmt->bindParam("login",$login);  
$stmt->bindParam("pass",$pass);  
$stmt->execute();
```

Modification

Update user

set login=:login, password=:pass

where login:=loginS

Suppression

Delete from user

Where login=:login

Un exemple plus complet

Le User possède des Post, un Post n'est possédé que par un user

Je vous écoute