BTS SIO :: SLAM 1

Date : Mai 2020

Utilisation avec PHP

Comment faire intéragir un programme PHP5 avec une base MySQL ?

Il existe trois possibilités appelées extensions (***connector***):

- mysql_ --> pour des version de MySQL antérieur à 4.1 (obsolète)
- mysqli_ --> i comme improved (amélioré)
- PDO (PHP Data Objects) --> couche d'abstraction orientée objet

Connexion à la base

Connaître la version de son MySQL :

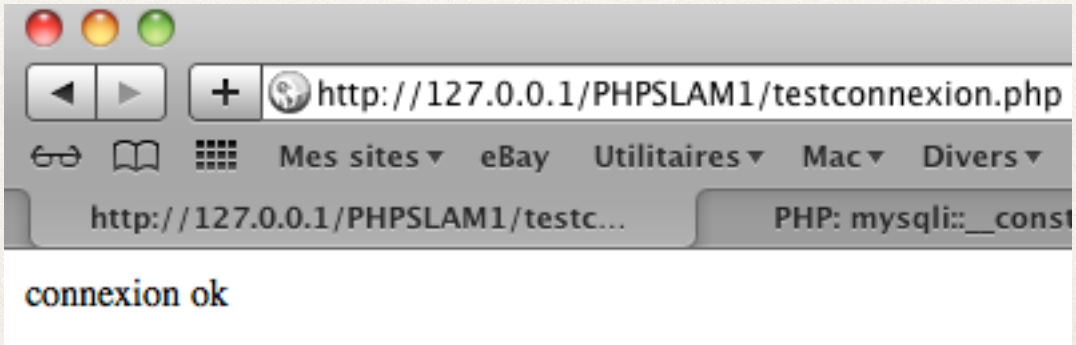
On va donc utiliser mysqli_ puisque notre serveur MySQL est > 4.1

La fonction pour se connecter est : mysqli_connect().
Cette fonction retourne un objet de connexion qu'on va réutiliser par la suite.
Si la connexion échoue la fonction renvoie FALSE.

Code pour la Connexion

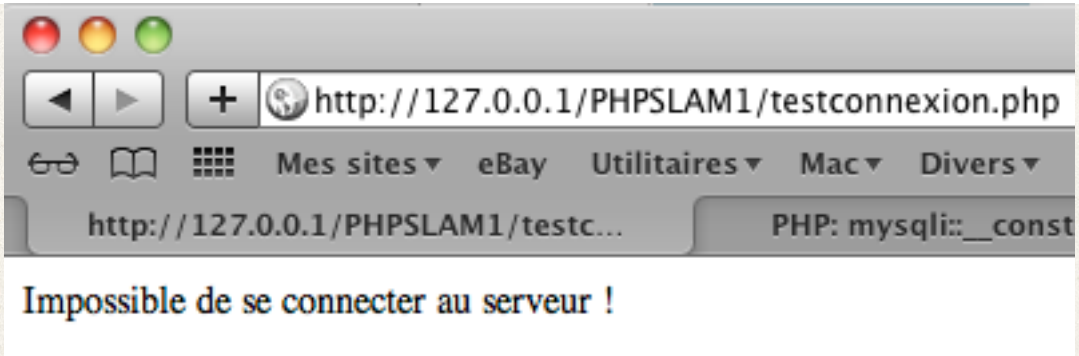
```php
<?php

// L'instruction pour se connecter est mysqli_connect
// Paramètre : host, user, mot de passe, [base], [port], [socket]
// Ici on précise la base : oiseaux mais pas le port , ni le socket
$service=mysqli_connect('localhost','root','root','oiseaux');
if ($service)
    {
        print "connexion ok";
        mysqli_close($service);
    }
else
    {
        print "Impossible de se connecter au serveur !";
    }
?>
```

Ce code est à placer dans votre espace web apache !



Essayons autre chose...

```php
<?php

// On introduit une erreur en mettant rout comme mot de passe
$service=mysqli_connect('localhost','root','rout','oiseaux');
if ($service)
    {
        print "connexion ok";
        mysqli_close($service);
    }
else
    {
        print "Impossible de se connecter au serveur !";
    }
?>
```



Ce qui est normal !

API de PHP pour MySQL

Interaction avec la base...

(Les paramètres ne sont pas indiqués)

- mysqli_connect : connexion
- mysqli_close : ferme la connexion dont l'identifiant est passé en paramètre.
- mysqli_select_db : permet de changer de base
- mysqli_change_user : modifie l'utilisateur.
- mysqli_query : exécute l'instruction SQL passé en paramètre

EXTRACTION plusieurs lignes

Lorsque la requête retourne plusieurs lignes (ce qui est souvent le cas...), il faut affecter le résultat dans un tableau.
On a besoin ensuite d'un **curseur** pour parcourir ce **tableau**.

Le curseur va permettre grâce à une boucle de parcourir toutes les lignes.
Une ligne est souvent un tableau aussi !

L'instruction est : **mysqli_fetch_array()**

Cette fonction retourne la ligne courante (donc un tableau) ou NULL si il n'y a plus de ligne disponible.

PASSONS A LA PRATIQUE

```php
<?php
$service=mysqli_connect('localhost','root','root','OISEAUX');
if ($service)
    {
        $requete="SELECT ORDRE, DIURNE FROM ORDRES;";// Requete SQL
        if ($resultat=mysqli_query($service,$requete))// Interogation
        {
        $ncols=mysqli_num_fields($resultat);// ncols pour savoir combien la ligne a de colonnes
        print "<TABLE BORDER=1> ";// On va placer les résultats dans un tableau
        while($ligne=mysqli_fetch_array($resultat))// On parcours chaque ligne
        {
            print "<TR> ";// Début de ligne en HTML <TR>
            for ($i=0;$i<$ncols;$i++)print "<TD> $ligne[$i] </TD>";
            print "</TR> ";// Fin de ligne en HTML
        }
        print "</TABLE>";// fin du tableau
        mysqli_close($service);
        }
        else print "<BR>La requete est un echec</BR>";

    }
else    print "Impossible de se connecter au serveur !";

?>
```

RESULTATS

ANSERIFORMES	
APODIFORMES	
BUCEROTIFORMES	
CAPRIMULGIFORMES	
CHARADRIIFORMES	
CICONIIFORMES	
COLIIFORMES	
COLUMBIFORMES	
CORACIIFORMES	
CUCULIFORMES	
FALCONIFORMES	1
GALLIFORMES	
GAVIIFORMES	
GRUIFORMES	
MUSOPHAGIFORMES	
PASSERIFORMES	
PELECANIFORMES	
PHOENICOPTERIFORMES	
PICIFORMES	
PODICIPEDIFORMES	

Code HTML

```
<TABLE BORDER=1> <TR> <TD> ANSERIFORMES </TD><TD> </TD></TR> <TR> <TD> APODIFORMES </TD><TD> </TD></TR> <TR> <TD>
BUCEROTIFORMES </TD><TD> </TD></TR> <TR> <TD> CAPRIMULGIFORMES </TD><TD> </TD></TR> <TR> <TD> CHARADRIIFORMES </TD
<TD> </TD></TR> <TR> <TD> CICONIIFORMES </TD><TD> </TD></TR> <TR> <TD> COLIIFORMES </TD><TD> </TD></TR> <TR> <TD>
COLUMBIFORMES </TD><TD> </TD></TR> <TR> <TD> CORACIIFORMES </TD><TD> </TD></TR> <TR> <TD> CUCULIFORMES </TD><TD>
</TD></TR> <TR> <TD> FALCONIFORMES </TD><TD> 1 </TD></TR> <TR> <TD> GALLIFORMES </TD><TD> </TD></TR> <TR> <TD>
GAVIIFORMES </TD><TD> </TD></TR> <TR> <TD> GRUIFORMES </TD><TD> </TD></TR> <TR> <TD> MUSOPHAGIFORMES </TD><TD>
</TD></TR> <TR> <TD> PASSERIFORMES </TD><TD> </TD></TR> <TR> <TD> PELECANIFORMES </TD><TD> </TD></TR> <TR> <TD>
PHOENICOPTERIFORMES </TD><TD> </TD></TR> <TR> <TD> PICIFORMES </TD><TD> </TD></TR> <TR> <TD> PODICIPEDIFORMES </TD
<TD> </TD></TR> <TR> <TD> PROCELLARIIFORMES </TD><TD> </TD></TR> <TR> <TD> PSITTACIFORMES </TD><TD> </TD></TR> <TR
<TD> PTEROCLIDIFORMES </TD><TD> </TD></TR> <TR> <TD> SPHENISCIFORMES </TD><TD> </TD></TR> <TR> <TD> STRIGIFORMES
</TD><TD> 0 </TD></TR> <TR> <TD> STRUTHIONIFORMES </TD><TD> </TD></TR> <TR> <TD> TINAMIFORMES </TD><TD> </TD></TR>
<TR> <TD> TROGONIFORMES </TD><TD> </TD></TR> <TR> <TD> TURNICIFORMES </TD><TD> </TD></TR> <TR> <TD> UPUPIFORMES
</TD><TD> </TD></TR> </TABLE>
```

Version objet

```php
<?php
    // Notation objet avec mysqli

    // création d'un objet de type connexion
    $connect = new mysqli('localhost','root','root','OISEAUX');
    if (!$connect->connect_errno) {
        // Récupération d'une instance avec la requête
        $results = $connect->query("SELECT ORDRE, DIURNE FROM ORDRES;");
        if ($results) {
            // On récupère le tableau
            while ($info = $results->fetch_array()) {
                echo 'Ordre :'.$info['ORDRE'].' type :'.$info['DIURNE'].'<br>';
            }

            $results->close();
        }
        else {
            echo 'Pas de résultats';
            $results->close();
        }
    }
    else {
        // Affiche le numéro d'erreur lors de la tentative de connexion
        echo 'Pas de connexion a MYSQL : Erreur N°'. $connect->connect_errno;
    }
    // Fermeture
    $connect->close();
?>
```

Interactions avec la base

La majorité des traitements SQL (lorsqu'ils incluent des paramètres) comporte 6 étapes :

- Connexion (connect)
- Préparation de l'ordre (parse)
- Association des paramètres à l'ordre SQL (bind)
- Exécution du dit ordre (execute)
- Lecture des lignes (fetch)
- Libération des ressources (free et close)

Interactions avec la base

Préparation, exécution

- mysqli_prepare : prépare l'ordre SQL retourne un objet qui peut être utilisé par mysqli_stmt_bind_param et mysqli_stmt_execute
- mysqli_stmt_execute : exécute l'ordre SQL préparé par mysqli_prepare
- mysqli_stmt_fetch : exécute pas à pas une requête préparée.
- mysqli_stmt_bind_result : pour associer des colonnes d'un résultat à des variables
- mysqli_stmt_bind_param : fait l'inverse de PHP vers MySQL

Interactions avec la base

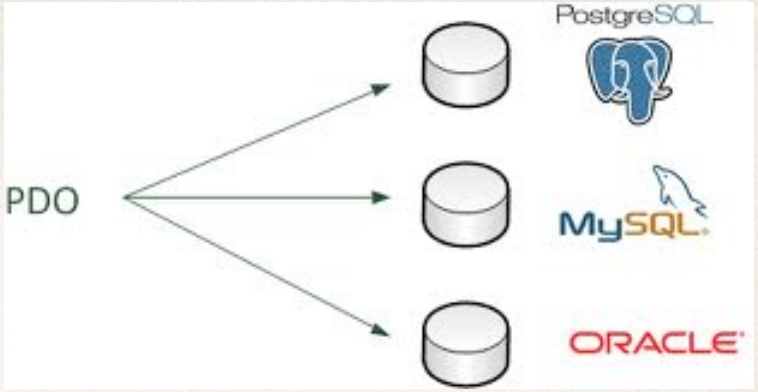
```php
<?php
// Connect
$service=mysqli_connect('localhost','root','root','OISEAUX');
if ($service)
    {
        $requete="SELECT ORDRE, DIURNE FROM ORDRES;";
        $ordre=mysqli_prepare($service,$requete);// parse
        if (($resultat=mysqli_stmt_execute($ordre))>0)// execute
        {
        mysqli_stmt_bind_result($ordre,$ord,$dir);// bind
        print "<TABLE BORDER=1> ";
        while(mysqli_stmt_fetch($ordre))// fetch
                    print "<TR><TD> $ord</TD><TD> $dir</TD></TR>";
        print "</TABLE>";
        mysqli_stmt_close($ordre);// free et close
        mysqli_close($service);
        }
        else print "<BR>La requete est un echec</BR>";

    }
else    print "Impossible de se connecter au serveur !";

?>
```

Les fonctions mysqli_stmt_bind_param et mysqli_stmt_bind_result permettent d'associer à des colonnes MySQL des variables PHP et inversement.

Ajout dans la base

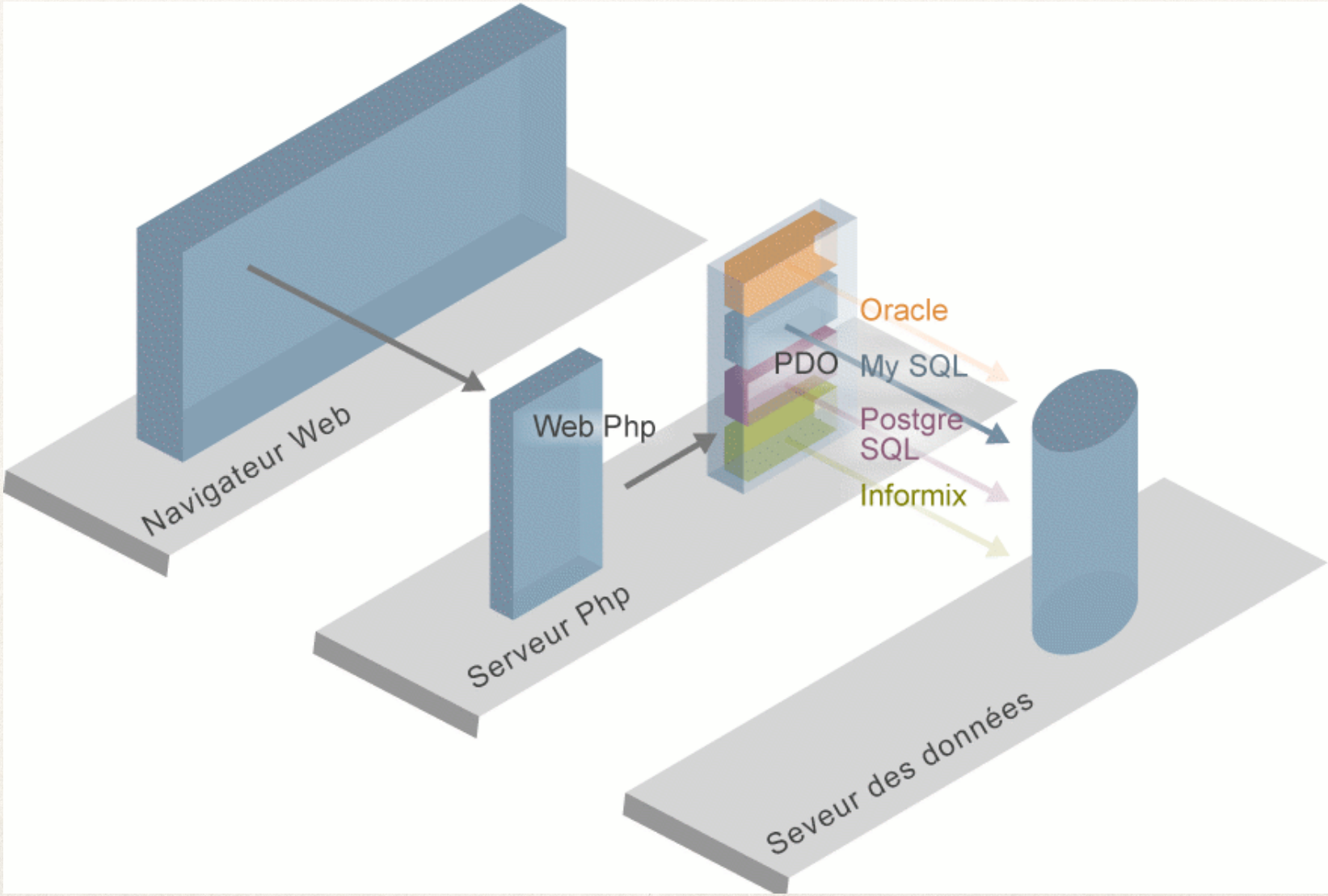
```php
<?php
// Connect
$service=mysqli_connect('localhost','root','root','AVIONS');
if ($service)
    {
        $requete="INSERT INTO AVIONS.Compagnie VALUES ('AL','Air Lib')";
        $ordre=mysqli_prepare($service,$requete);// parse
        if (($resultat=mysqli_stmt_execute($ordre))>0)// execute
            print "<BR> Ajout opéré";
        else
            print "<BR> Erreur";

        mysqli_stmt_close($ordre);// free et close
        mysqli_close($service);
    }
else    print "Impossible de se connecter au serveur !";

?>
```

Avec PDO



- PDO est une API faite pour fonctionner avec plusieurs SGBD différents.
- Elle offre donc une abstraction mais peut parfois être plus limité que les API spécifique au SGBD.
- Il existe 3 classes :
  - **PDO** : lien à la base de données
  - **PDOStatement** pour les requêtes
  - **PDOException** pour traiter les erreurs

Avec PDO


Navigateur Web
Web Php
Serveur Php
PDO
Oracle
My SQL
Postgre SQL
Informix
Seveur des données

PDO accès à la base de données

```php
<?php
    // Définition des variables de connexion
    $user = 'root';
    $password = 'root';
    $dsn = 'mysql:host=localhost;dbname=FILMS;charset=utf8';

    // Connexion à la base de données
    try {
        $db = new PDO($dsn,$user,$password);
    } catch (Exception $e) {
        die('Erreur : '. $e->getMessage());
    }
?>
```

Un DSN décrit la base de données à laquelle vous souhaitez accéder.

Syntaxe :

nom du pilote qu'on utilise : oci, mysql, etc.

Les autres dépendent du SGBDR ici : host (adresse du serveur), dbname (nom de la base de données), port (données facultatives)

PDO et requêtes SQL - Lecture

```php
<?php
    // Définition des variables de connexion
    $user = 'root';
    $password = 'root';
    $dsn = 'mysql:host=localhost;dbname=FILMS;charset=utf8';

    // Connexion à la base de données
    try {
        $db = new PDO($dsn,$user,$password);
    } catch (Exception $e) {
        die('Erreur : '. $e->getMessage());
    }

    // Lecture de la table
    $resultat = $db->query('SELECT nom FROM Acteur;');
    // On affiche les résultats un à un
    while ($donnees = $resultat->fetch())
    {
        echo $donnees['nom']."<br/>";
    }

    // fermeture du curseur
    $resultat->closeCursor();
?>
```

PDO/SQL - Lecture avec HTML

```php
<?php
    // Définition des variables de connexion
    $user = 'root';
    $password = 'root';
    $dsn = 'mysql:host=localhost;dbname=FILMS;charset=utf8';

    // Connexion à la base de données
    try {
        $db = new PDO($dsn,$user,$password);
    } catch (Exception $e) {
        die('Erreur : '. $e->getMessage());
    }

    // Lecture de la table
    $resultat = $db->query('SELECT * FROM Acteur;');
    // On affiche les résultats un à un
    while ($donnees = $resultat->fetch())
    {
    ?>
        <p>
            <strong>Nom </strong> : <?php echo $donnees['nom'] ?>
            <br/>
            <strong>Prénom </strong> : <?php echo $donnees['prenom'] ?>
            <br/>
        </p>
    <?php
    }

    // fermeture du curseur
    $resultat->closeCursor();
?>
```

PDO/SQL - Récupérer des variables

```php
<?php
    // Définition des variables de connexion
    $user = 'root';
    $password = 'root';
    $dsn = 'mysql:host=localhost;dbname=FILMS;charset=utf8';

    // Connexion à la base de données
    try {
        $db = new PDO($dsn,$user,$password);
    } catch (Exception $e) {
        die('Erreur : '. $e->getMessage());
    }
    ?>
    <form method="post" action="">
        <input type="text" name="type"/>
        <input type="submit" value="envoyer"/>
    </form>

    <?php
    if(isset($_POST['type']))
    {
        $req = $db->prepare('INSERT INTO TypesFilm (TypesFilmcol) VALUES (:mtype)');
        $req->execute(array('mtype' => $_POST['type']));
    }
?>
```