

PHP et MySQL



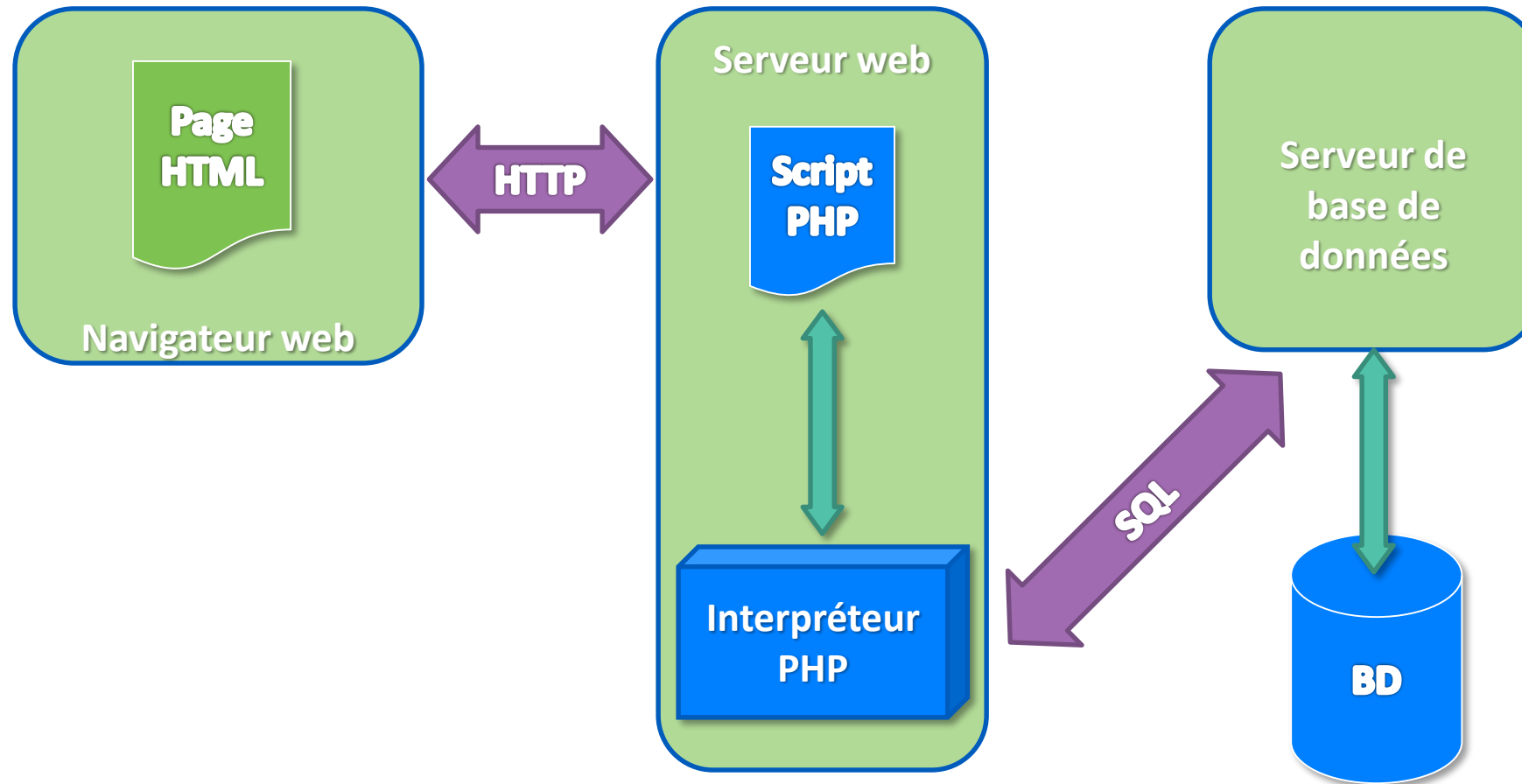
Institut Universitaire
de Technologie
Aix★Marseille Université

Introduction

Nous nous intéressons ici aux API (Applications Programming Interface) proposées par php pour les échanges entre php et les SGBD MySQL et PostgreSQL.

Nous n'aborderons que les API qui propose une syntaxe POO

introduction



INTERFACE PHP POUR MYSQL

LA CLASSE MYSQLI



Institut Universitaire
de Technologie

Aix★Marseille Université

La classe mysqli : connexion

Connexion au serveur:

```
$connexion=new mysqli("serveur", "user", "password", "database");
```

Exemple :

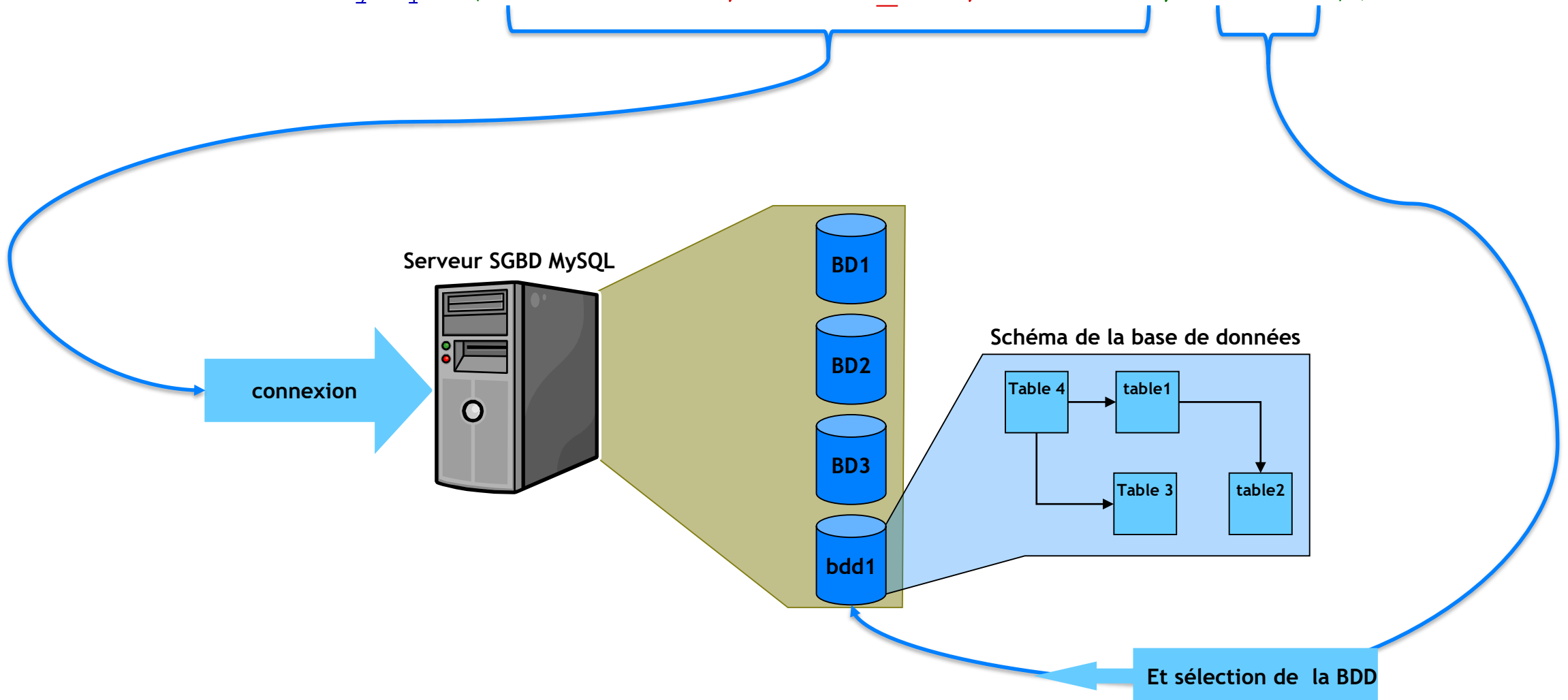
```
$connexion = new mysqli('localhost', 'user_bd','fdffsd', 'bdd1');
```

La base de données exemple

lcarmignac_php_iut T_UTILISATEUR_UTI	
UTI_LOGIN	: varchar(255)
UTI_PWD	: varchar(255)
UTI_MAIL	: varchar(255)
UTI_TEL	: varchar(20)
UTI_NOM	: varchar(255)
UTI_PRENOM	: varchar(255)
UTI_ADMIN	: tinyint(1)

La classe mysqli : connexion

```
$connexion = new mysqli('localhost', 'user_bd', 'fdffsd', 'bdd1');
```



La classe mysqli : fermeture de la connexion

Fermer la connexion

```
$connexion->close();
```

: permet de fermer la connexion à un serveur de bases de données

A noter que toutes les connexions aux serveurs de bases de données sont automatiquement fermées à la fin de l'exécution du script qui les aura ouvertes.

La classe mysqli : connexion

Exemple avec gestion des erreurs OBLIGATOIRE !!!

```
<?php
echo "test mysqli<br/>";
$connexion = new mysqli('localhost', 'user_bd', 'fdffsd', 'bdd1');

if ($connexion->connect_error) {
    echo('Erreur de connexion (' . $connexion->connect_errno . ') '
        . $connexion->connect_error);
}
echo "connexion réussie" ;
$connexion->close();
?>
```


La classe mysqli : Création et exécution d'une requête

Exécuter des requêtes avec la méthode query de la classe mysqli

La fonction **mysqli_query()** est la plus commune, et combine l'exécution de la requête avec une récupération de son jeu de résultats en mémoire tampon

Exemple (on suppose la connexion établit) :

```
$res = $connexion->query("SELECT * FROM T_UTILISATEUR_UTI ORDER BY UTI_NOM");  
if (!$res)  
    die ('Echec lors de l'exécution de la requête: ('. $ connexion->errno .') '  
        . $ connexion->error);  
  
echo "affiche le résultat avec un for... <br/>";  
for ($row_no = 0; $row_no <= $res->num_rows - 1; $row_no++) {  
    $res->data_seek($row_no);  
    $row = $res->fetch_assoc();  
    echo "    login = " . $row['UTI_LOGIN'] . " ; \n  
        mail = " . $row['UTI_MAIL'] . " ; \n  
        tel = " . $row['UTI_TEL'] . " ; \n  
        nom = " . $row['UTI_NOM'] . " ; \n  
        prenom = " . $row['UTI_PRENOM'] . " ; \n<br/>";  
}
```

La classe mysqli : Création et exécution d'une requête

Exécuter des requêtes avec la méthode query de la classe mysqli

La fonction **mysqli_query()** est la plus commune, et combine l'exécution de la requête avec une récupération de son jeu de résultats en mémoire tampon

Exemple (on suppose la connexion établit) :

```
$res = $connexion->query("SELECT * FROM T_UTILISATEUR_UTI ORDER BY UTI_NOM");  
if (!$res)  
    die ('Echec lors de l'exécution de la requête: ('. $ connexion->errno .') '  
        . $ connexion->error);
```

```
echo "affiche le résultat avec un for... <br/>";  
for ($row_no = 0; $row_no <= $res->num_rows - 1; $row_no++) {  
    $res->data_seek($row_no);  
    $row = $res->fetch_assoc();  
    echo "    login = " . $row['UTI_LOGIN'] . " ; \n  
        mail = " . $row['UTI_MAIL'] . " ; \n  
        tel = " . $row['UTI_TEL'] . " ; \n  
        nom = " . $row['UTI_NOM'] . " ; \n  
        prenom = " . $row['UTI_PRENOM'] . " ; \n<br/>";  
}
```

Exécute la requête
avec la gestion des erreurs

La classe mysqli : Création et exécution d'une requête

Exécuter des requêtes avec la méthode query de la classe mysqli

La fonction **mysqli_query()** est la plus commune, et combine l'exécution de la requête avec une récupération de son jeu de résultats en mémoire tampon

Exemple (on suppose la connexion établit) :

```
$res = $connexion->query("SELECT * FROM T_UTILISATEUR_UTI ORDER BY UTI_NOM");  
if (!$res)  
    die ('Echec lors de l'exécution de la requête: ('. $ connexion->errno .') '  
        . $ connexion->error);
```

```
echo "affiche le résultat avec un for... <br/>";  
for ($row_no = 0; $row_no <= $res->num_rows - 1; $row_no++) {  
    $res->data_seek($row_no);  
    $row = $res->fetch_assoc();  
    echo "    login = " . $row['UTI_LOGIN'] . " ; \n  
        mail = " . $row['UTI_MAIL'] . " ; \n  
        tel = " . $row['UTI_TEL'] . " ; \n  
        nom = " . $row['UTI_NOM'] . " ; \n  
        prenom = " . $row['UTI_PRENOM'] . " ; \n<br/>";  
}
```

Positionne le résultat à la ligne courante

Parcours du résultat avec
une boucle FOR

La classe mysqli : Création et exécution d'une requête

Exécuter des requêtes avec la méthode query de la classe mysqli

La fonction **mysqli_query()** est la plus commune, et combine l'exécution de la requête avec une récupération de son jeu de résultats en mémoire tampon

Exemple (on suppose la connexion établit) :

```
$res = $connexion->query("SELECT * FROM T_UTILISATEUR_UTI ORDER BY UTI_NOM");  
if (!$res)  
    die ('Echec lors de l'exécution de la requête: ('. $ connexion->errno .') '  
        . $ connexion->error);
```

```
echo "affiche le résultat avec un for... <br/>";  
for ($row_no = 0; $row_no <= $res->num_rows - 1; $row_no++) {  
    $res->data_seek($row_no);  
    $row = $res->fetch_assoc();  
    echo "    login = " . $row['UTI_LOGIN'] . " ; \n  
        mail = " . $row['UTI_MAIL'] . " ; \n  
        tel = " . $row['UTI_TEL'] . " ; \n  
        nom = " . $row['UTI_NOM'] . " ; \n  
        prenom = " . $row['UTI_PRENOM'] . " ; \n<br/>";  
}
```

Retourne la ligne courante sous la forme d'un tableau associatif

Parcours du résultat avec une boucle FOR

La classe mysqli : Création et exécution d'une requête

Exécuter des requêtes avec la méthode query de la classe mysqli

La fonction **mysqli_query()** est la plus commune, et combine l'exécution de la requête avec une récupération de son jeu de résultats en mémoire tampon

Exemple (on suppose la connexion établit) :

```
$res = $connexion->query("SELECT * FROM T_UTILISATEUR_UTI ORDER BY UTI_NOM");  
if (!$res)  
    die ('Echec lors de l'exécution de la requête: ('. $connexion->errno .') '  
        . $connexion->error);  
  
echo "affiche le résultat avec un for... <br/>";  
for ($row_no = 0; $row_no <= $res->num_rows - 1; $row_no++) {  
    $res->data_seek($row_no);  
    $row = $res->fetch_assoc();  
    echo "    login = " . $row['UTI_LOGIN'] . " ; \n  
        mail = " . $row['UTI_MAIL'] . " ; \n  
        tel = " . $row['UTI_TEL'] . " ; \n  
        nom = " . $row['UTI_NOM'] . " ; \n  
        prenom = " . $row['UTI_P  
}  
affiche le résultat avec un for..  
login = lolo ; mail = lolo@ff.fr ; tel = ; nom = carmi ; prenom = laurent ;  
login = test_user ; mail = test_user@gmail.fr ; tel = ; nom = Carmi ; prenom = Laurent ;
```

La classe mysqli : Création et exécution d'une requête

Exécuter des requêtes avec la méthode query de la classe mysqli

La fonction **mysqli_query()** est la plus commune, et combine l'exécution de la requête avec une récupération de son jeu de résultats en mémoire tampon

Exemple avec parcours avec une un WHILE(on suppose la connexion établit) :

```
$res = $connexion->query("SELECT * FROM T_UTILISATEUR_UTI ORDER BY UTI_NOM");  
if (!$res)  
    die ('Echec lors de l/'exécution de la requête: ('. $ connexion- >errno .') '  
                                                . $ connexion- >error);  
  
$res->data_seek(0);  
echo "<br/>affiche le résultat avec un while... <br/>";  
while ($row = $res->fetch_assoc()) {  
    echo "    login = " . $row['UTI_LOGIN'] . " ; \n  
          mail = " . $row['UTI_MAIL'] . " ; \n  
          tel = " . $row['UTI_TEL'] . " ; \n  
          nom = " . $row['UTI_NOM'] . " ; \n  
          prenom = " . $row['UTI_PRENOM'] . " ; \n<br/>";  
}
```

Positionne le résultat à la première ligne

La classe mysqli : Création et exécution d'une requête

Exécuter des requêtes avec la méthode query de la classe mysqli

La fonction **mysqli_query()** est la plus commune, et combine l'exécution de la requête avec une récupération de son jeu de résultats en mémoire tampon

Exemple avec parcours avec une un WHILE(on suppose la connexion établit) :

```
$res = $connexion->query("SELECT * FROM T_UTILISATEUR_UTI ORDER BY UTI_NOM");  
if (!$res)  
    die ('Echec lors de l'exécution de la requête: ('. $ connexion- >errno .') '  
        . $ connexion- >error);  
  
$res->data_seek(0);  
echo "<br/>affiche le résultat avec un while... <br/>";  
while ($row = $res->fetch_assoc()) {  
    echo "    login = " . $row['UTI_LOGIN'] . " ; \n  
        mail = " . $row['UTI_MAIL'] . " ; \n  
        tel = " . $row['UTI_TEL'] . " ; \n  
        nom = " . $row['UTI_NOM'] . " ; \n  
        prenom = " . $row['UTI_PRENOM'] . " ; \n<br/>";  
}
```

Retourne les lignes du
résultat

La classe mysqli : Création et exécution d'une requête

Exécuter des requêtes avec la méthode query de la classe mysqli :

La fonction **mysqli_query()** est la plus commune, et combine l'exécution de la requête avec une récupération de son jeu de résultats en mémoire tampon

Il existe d'autre moyen de récupérer le résultat :

```
$tab=$res->fetch_all(MYSQLI_ASSOC);
```

```
// retourne le résultat sous forme d'un tableau associatif  
// Il vous suffit ensuite de parcourir ce tableau
```

```
foreach ($tab as $row ) {  
    echo "  login = " . $row['UTI_LOGIN'] . " ; \n  
    mail = " . $row['UTI_MAIL'] . " ; \n  
    tel = " . $row['UTI_TEL'] . " ; \n  
    nom = " . $row['UTI_NOM'] . " ; \n  
    prenom = " . $row['UTI_PRENOM'] . " ;  
    \n<br/>";  
}
```

```
[0] {  ["UTI_LOGIN"]=>"lolo",  
        ["UTI_PWD"]=>"ssdqsd",  
        ["UTI_MAIL"]=>"lolo@ff.fr",  
        ["UTI_TEL"]=>NULL,  
        ["UTI_NOM"]=>"carmi« ,  
        ...  
      }  
[1] {  ["UTI_LOGIN"]=>"test_user"  
        ["UTI_PWD"]=>"trryer",  
        ["UTI_MAIL"]=>test user@gmail.fr  
        ["UTI_TEL"]=>NULL,  
        ...  
      }
```


La classe mysqli : pour aller plus loin

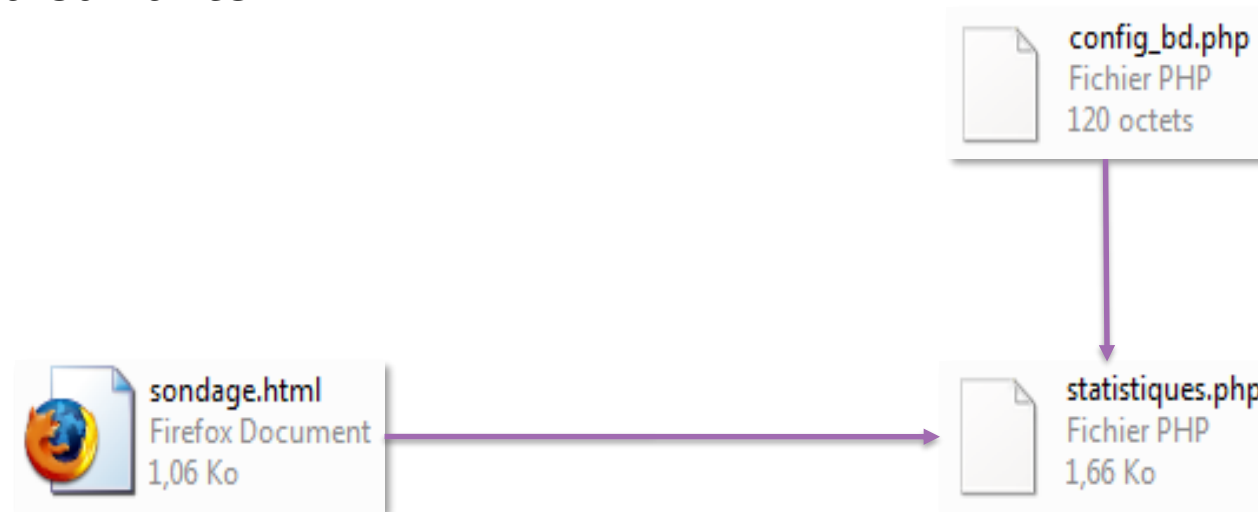
Et en savoir plus sur l'API mysqli

<http://php.net/manual/fr/book.mysqli.php>

Création d'un sondage très basique

Dans cet exemple, on utilise une base de données pour stocker les informations relatives au sondage.

L'architecture est la suivante :



Le formulaire : sondage.html

```
<!doctype html>
<html lang="fr">
<head>
    <title>Sondage</title>
    <meta charset="utf-8">
</head>

<body>
    <h2>Combien avez-vous d'animaux domestiques ?</h2>
    <form method="post" action="sondage.php">
        <p><input type="radio" name="choix" value="Aucune" checked="checked" />Aucun<br />
        <input type="radio" name="choix" value="1" />1<br />
        <input type="radio" name="choix" value="2" />2<br />
        <input type="radio" name="choix" value="3" />3<br />
        <input type="radio" name="choix" value="plus" />Plus<br /><br />

        Merci de laisser votre pseudo : <input type="text" name="pseudo" /><br />
        <input type="submit" name="form_sondage" value="Envoyer" />
        <input type="reset" name="form_sondage" value="Effacer" /></p>
    </form>
</body>
</html>
```

Combien avez-vous d'animaux domestiques ?

☐ Aucun
☐ 1
☒ 2
☐ 3
☐ Plus

Merci de laisser votre pseudo :

La base de données

Paramètres de connexion :

```
1 <?php
2 $serveur = "lcarmignac";
3 $user = "u_sondage";
4 $pw = "xxxxxxx";
5 $base = "sondage";
6
7 //Connection à la base de données
8 $connexion=new mysqli($serveur,$user,$pw,$base );
9 if ($connexion->connect_error) {
10     die('Erreur de connexion (' . $connexion->connect_errno . ') ' . $connexion->connect_error);
11 }
12 ?>
```

La table dans PhpMyAdmin

v	larmignac_sondage sondage
	nom : varchar(20)
	resultat : varchar(20)
	date : timestamp

Affichage de la base : statistiques.php

```

1 <?php
2 //connexion à la bdd
3 include_once 'config_bd.php';
4 if (isset($_REQUEST['form_sondage'])) {
5     //Ecriture et execution de la requête
6     $request = "INSERT INTO sondage (nom, resultat) VALUES('".$_REQUEST['pseudo']."','".$_REQUEST['choix']."'");
7     $res = $connexion->query($request);
8     if (!$res)
9         die ('Echec lors de l\'exécution de la requête: ('.$connexion->errno .') '.$connexion->error);
10 }
11
12 if (!$res = $connexion->query( query: "SELECT * FROM sondage"))
13     die ('Echec lors de l\'exécution de la requête: ('.$connexion->errno .') '.$connexion->error);
14
15 //Traitement de la requête
16 $sHtml='<table>
17     <td><b>Pseudo</b></td><td><b>Réponse</b></td><td><b>Date</b></td>';
18 while ($tuple = $res->fetch_assoc()) {
19     $sHtml.= "<tr><td>".$tuple["nom"]."</td>
20         <td>".$tuple["resultat"]."</td>
21         <td>".$tuple["date"]."</td>
22     </tr>";
23 }
24 $sHtml.="</table>";
25 $connexion->close();
26
27 <?>
28 <!doctype html>
29 <html lang="fr">
30 <head>
31     <title>Réponses</title>
32     <meta charset="utf-8">
33     <meta http-equiv="Content-Style-Type" content="text/css" />
34 </head>
35 <body>
36     <?php echo $sHtml ?>
37 </body>
38 </html>

```

Si on arrive bien du formulaire

Combien avez-vous d'animaux domestiques ?

- ☐ Aucun
☐ 1
☒ 2
☐ 3
☐ Plus

Merci de laisser votre pseudo : Luna

Envoyer Effacer

Insertion du résultat du sondage
dans la BDD

Affichage de la base : statistiques.php

```
1 <?php
2 //connexion à la bdd
3 include_once 'config_bd.php';
4 if (isset($_REQUEST['form_sondage'])) {
5     //Ecriture et execution de la requête
6     $request = "INSERT INTO sondage (nom, resultat) VALUES('".$_REQUEST['pseudo']."' , '".$_REQUEST['choix']."' )";
7     $res = $connexion->query($request);
8     if (!$res)
9         die ('Echec lors de l\'exécution de la requête: ('.$connexion->errno .') ' . $connexion->error);
10 }
11
12 if (!$res = $connexion->query( query: "SELECT * FROM sondage"))
13     die ('Echec lors de l\'exécution de la requête: ('.$connexion->errno .') ' . $connexion->error);
14
15 //Traitement de la requête
16 $sHtml='<table>
17     <tr><th><b>Pseudo</b></th><th><b>Réponse</b></th><th><b>Date</b></th></tr>';
18 while ($tuple = $res->fetch_assoc()) {
19     $sHtml.= "<tr><td>".$tuple["nom"]."</td>
20         <td>".$tuple["resultat"]."</td>
21         <td>".$tuple["date"]."</td>
22     </tr>";
23 }
24 $sHtml.="</table>";
25 $connexion->close();
26
27 <?>
28 <!doctype html>
29 <html lang="fr">
30 <head>
31     <title>Réponses</title>
32     <meta charset="utf-8">
33     <meta http-equiv="Content-Style-Type" content="text/css" />
34 </head>
35 <body>
36     <?php echo $sHtml ?>
37 </body>
38 </html>
```

Construction du tableau html
qui reflète le contenu de a BDD

Affichage de la base : statistiques.php

```

1 <?php
2 //connexion à la bdd
3 include_once 'config_bd.php';
4 if (isset($_REQUEST['form_sondage'])) {
5     //Ecriture et execution de la requête
6     $request = "INSERT INTO sondage (nom, resultat) VALUES('".$_REQUEST['pseudo']."'".$_REQUEST['choix']."'");
7     $res = $connexion->query($request);
8     if (!$res)
9         die ('Echec lors de l\'exécution de la requête: ('.$connexion->errno .') '.$connexion->error);
10 }
11
12 if (!$res = $connexion->query( query: "SELECT * FROM sondage"))
13     die ('Echec lors de l\'exécution de la requête: ('.$connexion->errno .') '.$connexion->error);
14
15 //Traitement de la requête
16 $sHtml='<table>
17     <td><b>Pseudo</b></td><td><b>Réponse</b></td><td><b>Date</b></td>';
18 while ($tuple = $res->fetch_assoc()) {
19     $sHtml.= "<tr><td>".$tuple["nom"]."</td>
20         <td>".$tuple["resultat"]."</td>
21         <td>".$tuple["date"]."</td>
22     </tr>";
23 }
24 $sHtml.="</table>";
25 $connexion->close();
26
27 <?>
28 <!doctype html>
29 <html lang="fr">
30 <head>
31     <title>Réponses</title>
32     <meta charset="utf-8">
33     <meta http-equiv="Content-Style-Type" content="text/css" />
34 </head>
35 <body>
36     <?php echo $sHtml ?>
37 </body>
38 </html>

```

Affiche le html généré

Pseudo	Réponse	Date
brett	1	2007-12-03 09:37:20
Martin	2	2007-12-03 09:37:28
Georges	1	2007-12-03 09:37:40
Anaïs	Acun	2007-12-03 09:37:49
John	Plus de 3	2007-12-03 09:37:59
Luna	2	2007-12-03 09:38:31

LA CLASSE PDO DU PHP

INTERFACE D'ACCÈS AUX BDD



Institut Universitaire
de Technologie
Aix★Marseille Université

Présentation

Définition

- PDO (**P**HP **D**ata **O**bjects) l'accès aux BDD en faisant abstraction du moteur de SGBD utilisé.

Avantages

- Abstraction du moteur de SGBD utilisé (en théorie) ;
- Orienté Objet : les objets PDO et PDOStatement peuvent être étendus, il est donc possible de personnaliser une partie du comportement initial ;
- Exception : l'interface PDO utilisent des exceptions, il est donc tout à fait possible d'intégrer facilement un système de gestion des erreurs.

Présentation

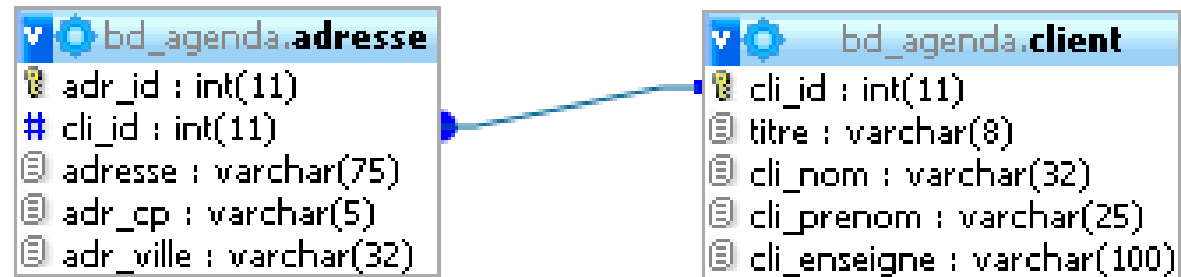
Installation

- Avec php5 l'extension (php_pdo.dll) n'est pas active, pour l'activer ouvrir le php.ini et décommentez la ligne `;extension=php_pdo.dll` en supprimant le point virgule `extension=php_pdo.dll`
- Cela n'est pas suffisant il faut aussi activer le driver* pour le SGBD utilisé, par exemple pour MySQL décommentez la ligne `extension=php_pdo_mysql.dll`

*Voir le tableau des DLL pour les SGBDs sur la [documentation officielle](#).

Avertissement

Schéma de données utilisé dans la suite



SGBD utilisé

- MySQL

Connexion

- ⦿ Création d'une instance de la classe PDO sans gestion des erreurs

```
<?php
// sans gestion d'erreur
//
define('DB_HOST', 'localhost'); // chemin vers le serveur
define('DB_PORT', '3307'); // port utilisé : 3306 par défaut, avec le portable web kit 3307
define('DB_DATABASE', 'bd_agenda'); // bdd utilisé
define('DB_USERNAME', 'u_bd_agenda'); // utilisateur autorisé
define('DB_PASSWORD', 'LNx8VLN'); // mot de passe de l'utilisateur

$PDO_BDD = new PDO('mysql:host='.DB_HOST.';port='.DB_PORT.';dbname='.DB_DATABASE, DB_USERNAME , DB_PASSWORD);
?>
```

Connexion

- ⦿ Création d'une instance de la classe PDO sans gestion des erreurs

```
<?php
// sans gestion d'erreur
//
define('DB_HOST', 'localhost'); // chemin vers le serveur
define('DB_PORT', '3307'); // port utilisé : 3306 par défaut, avec le portable web kit 3307
define('DB_DATABASE', 'bd_agenda'); // bdd utilisé
define('DB_USERNAME', 'u_bd_agenda'); // utilisateur autorisé
define('DB_PASSWORD', 'LNX8VLN'); // mot de passe de l'utilisateur

$PDO_BDD = new PDO('mysql:host='.DB_HOST.';port='.DB_PORT.';dbname='.DB_DATABASE,DB_USERNAME , DB_PASSWORD);
?>
```

Définition des constantes pour les paramètres de connexion

Connexion

- Création d'une instance de la classe PDO sans gestion des erreurs

```
<?php
// sans gestion d'erreur
//
define('DB_HOST', 'localhost'); // chemin vers le serveur
define('DB_PORT', '3307'); // port utilisé : 3306 par défaut, avec le portable web kit 3307
define('DB_DATABASE', 'bd_agenda'); // bdd utilisé
define('DB_USERNAME', 'u_bd_agenda'); // utilisateur autorisé
define('DB_PASSWORD', 'LNX8VLN'); // mot de passe de l'utilisateur

$PDO_BDD = new PDO('mysql:host='.DB_HOST.';port='.DB_PORT.';dbname='.DB_DATABASE,DB_USERNAME , DB_PASSWORD);
?>
```

Création d'une instance de la classe PDO avec les paramètres de connexion définis, le new appelle le constructeur de la classe PDO

Connexion avec gestion d'erreur

⊙ Connexion avec gestion des erreurs

```
<?php
// avec gestionnaire d'erreur
try
{
    $PDO_BDD = new PDO('mysql:host='.DB_HOST.';port='.DB_PORT.';dbname='.DB_DATABASE, DB_USERNAME , DB_PASSWORD);
    $PDO_BDD->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION); // rapport d'erreur : emit des exceptions
    $PDO_BDD->exec("SET NAMES 'utf8'"); // codage utilisé
}
catch(Exception $e) //interception de l'erreur
{
    echo 'Erreur : ' . $e->getMessage() . '<br />';
    echo 'N° : ' . $e->getCode();
    exit();
}
?>
```

La gestion des exceptions doit être gérée avec **try** et **catch**, tout code qui peut générer des exceptions doit être placé entre dans le **try**, une exception sera attrapée et traitée dans le **catch**

[Pour en savoir plus sur la gestion des exceptions avec php](#)

Connexion avec gestion d'erreur

⊙ Connexion avec gestion des erreurs

```
<?php
// avec gestionnaire d'erreur
try
{
    $PDO_BDD = new PDO('mysql:host='.DB_HOST.';port='.DB_PORT.';dbname='.DB_DATABASE, DB_USERNAME , DB_PASSWORD);
    $PDO_BDD->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION); // rapport d'erreur : emit des exceptions
    $PDO_BDD->exec("SET NAMES 'utf8'"); // codage utilisé
}
catch(Exception $e) //interception de l'erreur
{
    echo 'Erreur : ' . $e->getMessage() . '<br />';
    echo 'N° : ' . $e->getCode();
    exit();
}
?>
```

Création d'une instance de la classe PDO avec les paramètres de connexion définis et configuration de l'attribut* `PDO::ATTR_ERRMODE` du gestionnaire de base de données; `PDO::ERRMODE_EXCEPTION` précise que le mode d'erreur est géré par l'émission d'une exception

*Voir les attributs de la classe PDO

Connexion avec gestion d'erreur

⊙ Connexion avec gestion des erreurs

```
<?php
// avec gestionnaire d'erreur
try
{
    $PDO_BDD = new PDO('mysql:host='.DB_HOST.';port='.DB_PORT.';dbname='.DB_DATABASE, DB_USERNAME , DB_PASSWORD);
    $PDO_BDD->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION); // rapport d'erreur :emet des exceptions
    $PDO_BDD->exec("SET NAMES 'utf8'"); // codage utilisé
}
catch(Exception $e) //interception de l'erreur
{
    echo 'Erreur : '. $e->getMessage(). '<br />';
    echo 'N° : '. $e->getCode();
    exit();
}
?>
```

Attrape l'erreur et crée une instance (\$e) de la classe *PDOException**, la manipulation des méthodes de cette instance permet l'affichage du message d'erreur et du N° de l'erreur. Il est bien évident qu'à ce niveau il faut stopper l'exécution du script php (exit())

*Voir les propriétés et méthodes de la classe [PDOException](#)

Connexion avec gestion d'erreur

- ⊙ Exemples d'erreur possible lors de la connexion
 - L'utilisateur n'existe pas, le mot de passe est incorrect

```
Erreur : SQLSTATE[28000] [1045] Access denied for user 'bd_agenda'@'localhost' (using password: YES)  
N° : 0
```

- Le serveur de base de données n'est pas en fonctionnement

```
Erreur : SQLSTATE[HY000] [2003] Can't connect to MySQL server on 'localhost' (10061)  
N° : 0
```

Exécuter une requête

Introduction

- ⊙ On distingue deux méthodes de la classe PDO permettant d'exécuter deux types de requête :
 - **La méthode query()** pour l'interrogation de données, dès que vous utilisez le **SELECT**
 - **La méthode exec()** pour la manipulation de données, dès que vous utilisez le **UPDATE, INSERT ou DELETE**
- ⊙ Pour faire simple :

une requête qui retourne un résultat doit être exécutée avec la méthode query()
dans les autres cas on utilise la méthode exec();

Exécuter une requête

Avec la méthode `exec()`

- ⦿ On souhaite modifier l'adresse du client N° 13

```
<?php
include 'config.BD.inc.php'; // connexion à la bdd
// exécution du update avec ma méthode exec()
try
{
    $iNb_ligne=$PDO_BDD->exec("UPDATE adresse
                                SET adresse='13 rue Voltaire'
                                WHERE cli_id=13");
}
catch(Exception $e) //interception de l'erreur
{
    die ('Erreur : ' . $e->getMessage() . '<br />');
}

echo '<div style="font-weight:bold">' . $iNb_ligne . ' ligne(s) modifiée(s)</div>';

?>
```

Il est important de gérer les éventuelles erreurs d'exécution SQL.
Comme lors de la connexion on utilise le **try** et **catch**

Exécuter une requête

Avec la méthode exec()

- ⦿ On souhaite modifier l'adresse du client N° 13

```
<?php
include 'config.BD.inc.php'; // connexion à la bdd
// exécution du update avec ma méthode exec()
try
{ $iNb_ligne=$PDO_BDD->exec("UPDATE adresse
                           SET adresse='13 rue Voltaire'
                           WHERE cli_id=13"); }
catch(Exception $e) //interception de l'erreur
{
    die ('Erreur : ' . $e->getMessage() . '<br />');
}

echo '<div style="font-weight:bold">' . $iNb_ligne . ' ligne(s) modifiée(s)</div>';

?>
```

Exécution de la requête de modification;

La méthode exec() retourne le nombre de lignes affectées par les requêtes de type DELETE, UPDATE ou INSERT

Exécuter une requête

Avec la méthode exec()

- On souhaite modifier l'adresse du client N° 13

```
<?php
include 'config.BD.inc.php'; // connexion à la bdd
// exécution du update avec ma méthode exec()
try
{
    $iNb_ligne=$PDO_BDD->exec("UPDATE adresse
                                SET adresse='13 rue Voltaire'
                                WHERE cli id=13");
}

catch(Exception $e) //interception de l'erreur
{
    die ('Erreur : ' . $e->getMessage() . '<br />');
}

echo '<div style="font-weight:bold">' . $iNb_ligne . ' ligne(s) modifiée(s)</div>';

?>
```

Interception de l'éventuelle erreur est affiche de celle-ci, le **die** affiche le message et stoppe l'exécution du script

Exemple: si la table invoquée dans la requête n'existe pas

```
Erreur : SQLSTATE[42S02]: Base table or view not found: 1146 Table 'bd_agenda.adresses' doesn't exist
```

Exécuter une requête

Avec la méthode `exec()`

- ⦿ On souhaite modifier l'adresse du client N° 13

```
<?php
include 'config.BD.inc.php'; // connexion à la bdd
// exécution du update avec ma méthode exec()
try
{
    $iNb_ligne=$PDO_BDD->exec("UPDATE adresse
                               SET adresse='13 rue Voltaire'
                               WHERE cli_id=13");
}
catch(Exception $e) //interception de l'erreur
{
    die ('Erreur : ' . $e->getMessage() . '<br />');
}

echo '<div style="font-weight:bold">' . $iNb_ligne . ' ligne(s) modifiée(s)</div>';

?>
```

Affiche le nombre de lignes affectées par la requête

Exécuter une requête

Avec la méthode query()

- On souhaite afficher le nom, le prénom et ville des clients

Exécution de la
requête et
traitement des
erreurs

traitement du
résultat

```
<?php
include 'config.BD.inc.php'; // connexion à la bdd
// texte de la requête
$$Texte_REQ='SELECT cli_nom,cli_prenom,adr_ville
              FROM client NATURAL JOIN adresse ';

try
{
    $$Resultat=$PDO_BDD->query($$Texte_REQ);
}
catch(Exception $e) //interception de l'erreur
{
    die ('<div style="font-weight:bold; font-color:red">Erreur : '.$e->getMessage().'</div>');
}

$$Tab=$$Resultat->fetchAll(PDO::FETCH_ASSOC); // retourne le résultat dans un tableau
echo '<table style="text-align:left"><tr><th>Nom</th><th>Prénom</th><th>Ville</th></tr>';
foreach ($$Tab as $tligne)
{
    echo '<tr>';
    foreach ( $tligne as $valeur)
        echo '<td>'.$valeur.'</td>';
    echo '</tr>';
}
echo '</table>';
?>
```

La méthode query() retourne un objet de la classe **PDOStatement** qui nous permet de traiter le résultat de la requête SELECT

Exécuter une requête

Avec la méthode query()

- ⦿ Comment traiter le résultat
 - Il existe plusieurs méthodes de récupération du résultat définies par les constantes **PDO::FETCH *** et plusieurs méthodes pour le parcourir
 - Dans notre exemple on utilise la méthode PDO::FETCH_ASSOC qui spécifie qu'une ligne du résultat sera traitée comme un tableau associatif;
 - La méthode -> fetchAll() de la classe **PDOStatement** retourne l'intégralité du résultat dans un tableau

```
$sTab=$sResultat->fetchAll(PDO::FETCH_ASSOC); // retourne le résultat dans un tableau
```

Traiter le résultat

Traiter le résultat retourné par la méthode query()

- Comment traiter le résultat
 - La méthode `->fetchAll()` de la classe PDOStatement retourne l'intégralité du résultat dans un tableau.

```
$sTab=$sResultat->fetchAll(PDO::FETCH_ASSOC); // retourne le résultat dans un tableau
```

- Il aura autant d'éléments que de lignes retournées par la requête

Contenu du tableau
\$sTab

```
Array
(
    [0] => Array
        (
            [cli_nom] => DREYFUS
            [cli_prenom] => Jean
            [adr_ville] => PARIS
        )

    [1] => Array
        (
            [cli_nom] => DUVAL
            [cli_prenom] => Arsène
            [adr_ville] => PARIS
        )
)
```

Traiter le résultat

- Il suffit donc de parcourir ce tableau pour afficher le résultat

```
$sTab=$sResultat->fetchAll(PDO::FETCH_ASSOC); // retourne le résultat dans un tableau

echo '<table style="text-align:left">
    <tr><th>Nom</th><th>Prénom</th><th>Ville</th></tr>';
foreach ($sTab as $tligne)
{
    echo '<tr>';
    foreach ( $tligne as $value)
        echo '<td>'.$value.'</td>';
    echo '</tr>';
}
echo '</table>';
```

```
Array
(
    [0] => Array
        (
            [cli_nom] => DREYFUS
            [cli_prenom] => Jean
            [adr_ville] => PARIS
        )

    [1] => Array
        (
            [cli_nom] => DUVAL
            [cli_prenom] => Arsène
            [adr_ville] => PARIS
        )
)
```

Affichage du résultat dans un tableau HTML

Nom	Prénom	Ville
DREYFUS	Jean	PARIS
DUVAL	Arsène	PARIS
PHILIPPE	André	PARIS
PIERRELAYE	Paul	PARIS
DAUMIER	Amélie	PARIS

Traiter le résultat

Autre exemple pour traiter le résultat d'une requête

- ⦿ Avec la méthode -> fetch()

```
<?php
include 'config.BD.inc.php'; // connexion à la bdd

// texte de la requête
$sTexte_REQ='SELECT cli_nom,cli_prenom,adr_ville
              FROM client NATURAL JOIN adresse ';

try
{ $sResultat=$PDO_BDD->query($sTexte_REQ); }
catch(Exception $e) //interception de l'erreur
{ die ('<div style="font-weight:bold; color:red">Erreur : '.$e->getMessage().'</div>'); }

echo '<table><tr><th>Nom</th><th>Prénom</th><th>Ville</th></tr>';
while ( $ligne = $sResultat->fetch(PDO::FETCH_ASSOC)) // chaque ligne est un tableau associatif
    echo '<tr>
        <td>'.$ligne['cli_nom'].'</td>
        <td>'.$ligne['cli_prenom'].'</td>
        <td>'.$ligne['adr_ville'].'</td>
    </tr>';

echo '</table>';
?>
```

Exécuter une requête "préparée"

Avec la méthode `prepare()`

⊙ Présentation

- Cette méthode prépare une requête SQL à être exécutée en offrant la possibilité de mettre des marqueurs qui seront substitués lors de l'exécution.

Il existe deux types de marqueurs qui sont respectivement ?

```
"SELECT cli_nom FROM client WHERE cli_id = ?"
```

et les **marqueurs nominatifs**.

```
"SELECT cli_nom FROM client WHERE cli_id = :id"
```

Exécuter une requête "préparée"

Différence entre requête normale et requête préparée

Schéma d'une requête normale :

- 1 : envoi de la requête par le client vers le SGBD
- 2 : compilation de la requête
- 3 : plan d'exécution par le SGBD
- 4 : exécution de la requête
- 5 : résultat du SGBD vers le client

Schéma d'une requête préparée :

Phase 1 :

- 1 : envoi de la requête à préparer
- 2 : compilation de la requête
- 3 : plan d'exécution par le SGBD
- 4 : stockage de la requête compilée en mémoire
- 5 : retour d'un id de requête au client

Phase 2 :

- 1 : le client demande l'exécution de la requête avec l'id
- 2 : exécution
- 3 : résultat du SGBD au client

Exécuter une requête "préparée"

Avec la méthode `prepare()`

⦿ Avantage

- Optimisation des performances pour des requêtes appelées plusieurs fois ;
- limiter la bande passante utilisée entre le client et le serveur : dû au fait que l'échange d'informations est limité au strict minimum, la requête est stockée en mémoire du SGBD.
- Protection des injections SQL;
 - L'attaque par SQL injection, consiste à injecter une requête SQL dans un paramètre. Si ce paramètre n'est pas contrôlé par l'application alors des actions non prévues peuvent être pilotées à distance par un pirate.

Exécuter une requête "préparée"

Avec la méthode `prepare()`

- ⊙ Exemple d'injection SQL

Exemple de code PHP/MySQL vulnérable à une SQL injection
<pre><?php <u>mysql_query</u>("SELECT * FROM utilisateurs WHERE login='\$login'"); ?></pre>
Code SQL injecté
<pre>toto'; DROP TABLE utilisateurs; #</pre>
Requête SQL exécutée par l'application
<pre>SELECT * FROM utilisateurs WHERE login='toto'; DROP TABLE utilisateurs; #'</pre>

Cet exemple d'injection supprime la table des utilisateurs dans la base de données. D'autres exemples permettent de forcer la validation d'un contrôle d'accès et de s'approprier les droits d'un utilisateur, d'afficher le contenu d'une base de données...

Exécuter une requête "préparée"

Avec la méthode `prepare()`

- Exemple sans marqueur: nom des clients

```
<?php
include 'config.BD.inc.php'; // connexion à la bdd

try
{
    $Req_prepare=$PDO_BDD->prepare('SELECT cli_nom FROM client');
    $Req_prepare->execute();
}

catch(Exception $e) //interception de l'erreur
{
    die ('<div style="font-weight:bold; color:red">Erreur : ' . $e->getMessage() . '</div>');
}

while($lignes=$Req_prepare->fetch(PDO::FETCH_NUM))
    echo $lignes[0].'<br />';
?>
```

Prépare (`$PDO_BDD->prepare()`)
et exécute (`$Req_prepare->execute();`) la requête

ATTENTION une requête préparée doit toujours être exécutée

Exécuter une requête "préparée"

Avec la méthode `prepare()`

- ⦿ Exemple sans marqueur: nom des clients

```
<?php
include 'config.BD.inc.php'; // connexion à la bdd

try
{
    $Req_prepare=$PDO_BDD->prepare('SELECT cli_nom FROM client');
    $Req_prepare->execute();
}

catch(Exception $e) //interception de l'erreur
{
    die ('<div style="font-weight:bold; color:red">Erreur : ' . $e->getMessage() . '</div>');
}

while($lignes=$Req_prepare->fetch(PDO::FETCH_NUM))
    echo $lignes[0].'<br />';
?>
```

Traitement des erreurs

Exécuter une requête "préparée"

Avec la méthode `prepare()`

- ⦿ Exemple sans marqueur: nom des clients

```
<?php
include 'config.BD.inc.php'; // connexion à la bdd

try
{
    $Req_prepare=$PDO_BDD->prepare('SELECT cli_nom FROM client');
    $Req_prepare->execute();
}
catch(Exception $e) //interception de l'erreur
{
    die('<div style="font-weight:bold; color:red">Erreur : '.$e->getMessage().'</div>');
}

while($lignes=$Req_prepare->fetch(PDO::FETCH_NUM))
    echo $lignes[0].'<br />';
?>
```

Traitement du résultat :

On voit ici l'utilisation d'une autre méthode de récupération des lignes de résultat `PDO::FETCH_NUM` qui spécifie que chaque ligne est un tableau indexé

Exécuter une requête "préparée"

Avec la méthode `prepare()`

◉ Exemple avec marqueur ?

- Nom des clients qui s'appelle Gérard et qui sont des particuliers

```
<?php
include 'config.BD.inc.php'; // connexion à la bdd
$PrenomCli = 'Gérard';
$Type = 'Particulier';

try
{
    $Req_prepare = $PDO_BDD->prepare("SELECT cli_nom, cli_prenom FROM client where cli_prenom=? AND cli_enseigne=?");
    $Req_prepare->execute(array($PrenomCli, $Type));
}

catch(Exception $e) // interception de l'erreur
{
    die('<div style="font-weight:bold; color:red">Erreur : ' . $e->getMessage() . '</div>');
}

while($lignes = $Req_prepare->fetch(PDO::FETCH_NUM))
    echo $lignes[0] . ' ' . $lignes[1] . '<br />';
?>
```

Prépare (`$PDO_BDD->prepare()`) la requête, les ? seront substitués par les valeurs passées lors de l'appelle de la méthode `execute()`

Exécuter une requête "préparée"

Avec la méthode prepare()

- ⦿ Exemple avec marqueur ?
 - Nom des clients qui s'appelle Gérard et qui sont des particuliers

```
<?php
include 'config.BD.inc.php'; // connexion à la bdd
$SsPrenomCli='Gérard';
$SsType='Particulier';
try
{ $Req_prepare=$PDO_BDD->prepare("SELECT cli nom,cli prenom FROM client where cli prenom=? AND cli enseigne=?");
  $Req_prepare->execute(array($SsPrenomCli,$SsType));
}
catch(Exception $e) //interception de l'erreur
{ die ('<div style="font-weight:bold; color:red">Erreur : '.$e->getMessage().'</div>'); }

while ($lignes=$Req_prepare->fetch(PDO::FETCH_NUM))
  echo $lignes[0].' '.$lignes[1].<br />';
?>
```

Exécute la requête, un tableau d'indice est passé comme paramètre à la méthode execute(), il doit comporter autant de valeur qu'il y a de marqueurs

Exécuter une requête "préparée"

Avec la méthode `prepare()`

- ⦿ Exemple avec marqueur ?
 - Nom des clients qui s'appelle Gérard et qui sont des particuliers

```
<?php
include 'config.BD.inc.php'; // connexion à la bdd
$PrenomCli='Gérard';
$Type='Particulier';
try
{
    $Req_prepare=$PDO_BDD->prepare("SELECT cli_nom,cli_prenom FROM client where cli_prenom=? AND cli_enseigne=?");
    $Req_prepare->execute(array($PrenomCli,$Type));
}
catch(Exception $e) //interception de l'erreur
{
    die ('<div style="font-weight:bold; color:red">Erreur : ' . $e->getMessage() . '</div>');
}

while ($lignes=$Req_prepare->fetch(PDO::FETCH_NUM))
    echo $lignes[0] . ' ' . $lignes[1] . '<br />';
?>
```

Traitement du résultat

Exécuter une requête "préparée"

Avec la méthode prepare()

- Exemple avec marqueur **nominatif** :
 - Nom des clients qui s'appelle Gérard et qui sont des particuliers

```
<?php
include 'config.BD.inc.php'; // connexion à la bdd
$PrenomCli='Gérard';
$Type='Particulier';
try
{
    $Req_prepare=$PDO_BDD->prepare("SELECT cli_nom,cli_prenom FROM client where cli_prenom=:nom AND cli_enseigne=:type");
    $Req_prepare->execute(array('nom'=>$PrenomCli,'type'=>$Type));
}
catch(Exception $e) //interception de l'erreur
{
    die ('<div style="font-weight:bold; color:red">Erreur : '.$e->getMessage().'</div>');
}

while($lignes=$Req_prepare->fetch(PDO::FETCH_OBJ))
    echo $lignes->cli_nom.' ' . $lignes->cli_prenom.'<br />';
?>
```

Prépare (`$PDO_BDD->prepare()`) la requête, les marqueurs nominatifs seront substitués par les valeurs passées lors de l'appelle de la méthode `execute()`

Exécuter une requête "préparée"

Avec la méthode prepare()

- ⦿ Exemple avec marqueur **nominatif** :
 - Nom des clients qui s'appelle Gérard et qui sont des particuliers

```
<?php
include 'config.BD.inc.php';// connexion à la bdd
$PrenomCli='Gérard';
$Type='Particulier';

try
{ $Req_prepare=$PDO_BDD->prepare("SELECT cli nom,cli prenom FROM client where cli prenom=:nom AND cli enseigne=:type");
  $Req_prepare->execute(array('nom'=>$PrenomCli,'type'=>$Type)); }
catch(Exception $e) //interception de l'erreur
{ die ('<div style="font-weight:bold; color:red">Erreur : '.$e->getMessage().'</div>'); }

while($lignes=$Req_prepare->fetch(PDO::FETCH_OBJ))
  echo $lignes->cli_nom.' ' . $lignes->cli_prenom.'<br />';

?>
```

Exécute la requête, un tableau associatif est passé comme paramètre à la méthode execute(), il doit comporter autant de couple clé, valeur qu'il y a de marqueurs nominatifs

Exécuter une requête "préparée"

Avec la méthode prepare()

- ⦿ Exemple avec marqueur **nominatif** :
 - Nom des clients qui s'appelle Gérard et qui sont des particuliers

```
<?php
include 'config.BD.inc.php'; // connexion à la bdd
$PrenomCli='Gérard';
$Type='Particulier';
try
{
    $Req_prepare=$PDO_BDD->prepare("SELECT cli_nom,cli_prenom FROM client where cli_prenom=:nom AND cli_enseigne=:type");
    $Req_prepare->execute(array('nom'=>$PrenomCli,'type'=>$Type));
}
catch(Exception $e) //interception de l'erreur
{
    die ('<div style="font-weight:bold; color:red">Erreur : '.$e->getMessage().'</div>');
}

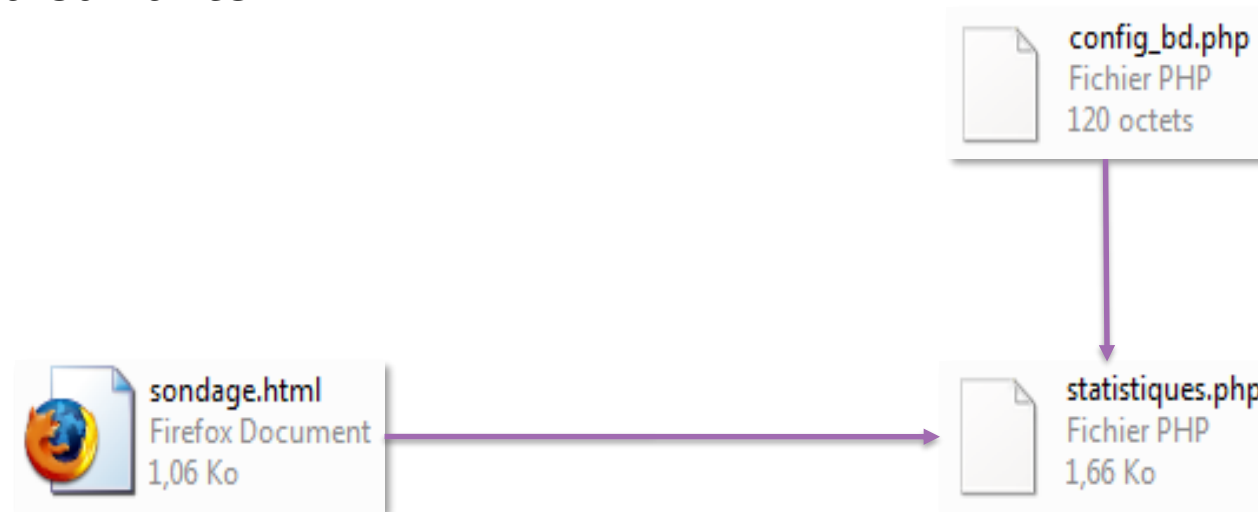
while($lignes=$Req_prepare->fetch(PDO::FETCH_OBJ))
    echo $lignes->cli_nom.' ' . $lignes->cli_prenom.'<br />';
?>
```

Traitement du résultat : On voit ici l'utilisation d'une autre méthode de récupération des lignes de résultat PDO::FETCH_OBJ qui spécifie que chaque ligne est un objet

Reprenons l'exemple de la création d'un sondage

Dans cet exemple, on utilise une base de données pour stocker les informations relatives au sondage.

L'architecture est la suivante :



Le formulaire : sondage.html

```
<!doctype html>
<html lang="fr">
<head>
    <title>Sondage</title>
    <meta charset="utf-8">
</head>

<body>
    <h2>Combien avez-vous d'animaux domestiques ?</h2>
    <form method="post" action="sondage.php">
        <p><input type="radio" name="choix" value="Aucune" checked="checked" />Aucun<br />
        <input type="radio" name="choix" value="1" />1<br />
        <input type="radio" name="choix" value="2" />2<br />
        <input type="radio" name="choix" value="3" />3<br />
        <input type="radio" name="choix" value="plus" />Plus<br /><br />

        Merci de laisser votre pseudo : <input type="text" name="pseudo" /><br />
        <input type="submit" name="form_sondage" value="Envoyer" />
        <input type="reset" name="form_sondage" value="Effacer" /></p>
    </form>
</body>
</html>
```

Combien avez-vous d'animaux domestiques ?

☐ Aucun
☐ 1
☒ 2
☐ 3
☐ Plus

Merci de laisser votre pseudo :

La base de données

Paramètres de connexion :

```
1 <?php
2 define ('SERVEUR', "lcarmignac");
3 define ('PORT', '3306');
4 define ('USER', "u_sondage");
5 define ('PW', "xxxxxx");
6 define ('BASE', "sondage");
7
8 //Connection à la base de données
9 try
10 {
11     $pdo_bdd=new PDO( dsn: 'mysql:host='.SERVEUR.';port='.PORT.' ;dbname='.BASE, username: USER, passwd: PW );
12     $pdo_bdd->setAttribute( attribute: PDO::ATTR_ERRMODE, value: PDO::ERRMODE_EXCEPTION );
13     $pdo_bdd->exec( statement: "SET NAMES 'utf8'" ); // codage utilisé lors des échanges entre le serveur et le client
14 }
15
16 catch (Exception $e)
17 {
18     echo 'Erreur de connexion: '.$e->getMessage().'\n';
19     echo 'N° : '.$e->getCode();
20     exit();
21 }
22 ?>
```

lcarmignac_sondage sondage
nom : varchar(20)
resultat : varchar(20)
date : timestamp

La table dans PhpMyAdmin

Affichage de la base : statistiques.php

```

1 <?php
2 //connexion à la bdd
3 include_once 'config_bd.php';
4 if (isset($_REQUEST['form_sondage'])) { //Ecriture et execution de la requete
5
6     $requete = "INSERT INTO sondage (nom, resultat) VALUES ('".$_REQUEST['pseudo']."','".$_REQUEST['choix']."')";
7     try
8     { $res = $pdo_bdd->query($requete); }
9     catch(Exception $e)
10     { die ('Echec lors de l\'exécution de la requête: ('.$e->getMessage() .'') '.$e->getCode()); }
11 }
12
13 try
14 { $res = $pdo_bdd->query( statement: "SELECT * FROM sondage" ); }
15 catch (Exception $e )
16 { die ('Echec lors de l\'exécution de la requête: ('.$e->getMessage() .'') '.$e->getCode()); }
17
18 //Traitement de la requête
19 $sHtml='<table>
20     <tr><th>Pseudo</th><th>Réponse</th><th>Date</th></tr>';
21 $qTab=$res->fetchAll( fetch_style: PDO::FETCH_ASSOC);
22 foreach ($qTab as $tuple) {
23     $sHtml.= "<tr><td>".$tuple["nom"]."</td>
24             <td>".$tuple["resultat"]."</td>
25             <td>".$tuple["date"]."</td>
26             </tr>";
27 }
28 $sHtml.="</table>";
29
30 <?>
31 <!doctype html>
32 <html lang="fr">
33 <head>
34     <title>Réponses</title>
35     <meta charset="utf-8">
36     <meta http-equiv="Content-Style-Type" content="text/css" />
37 </head>
38 <body>
39     <?php echo $sHtml ?>
40     <a href="sondage.html">retour vers le sondage</a>
41 </body>
42 </html>

```

Si on arrive bien du formulaire

Combien avez-vous d'animaux domestiques ?

- ☐ Aucun
☐ 1
☒ 2
☐ 3
☐ Plus

Merci de laisser votre pseudo : Luna

Insertion du résultat du sondage dans la BDD

Affichage de la base : statistiques.php

```

1 <?php
2 //connexion à la bdd
3 include_once 'config_bd.php';
4 if (isset($_REQUEST['form_sondage'])) { //Ecriture et execution de la requete
5
6     $requete = "INSERT INTO sondage (nom, resultat) VALUES ('".$_REQUEST['pseudo']."','".$_REQUEST['choix']."')";
7     try
8     {
9         $res = $pdo_bdd->query($requete);
10    }
11    catch(Exception $e)
12    {
13        die ('Echec lors de l\'exécution de la requête: ('.$e->getMessage() .') ' . $e->getCode());
14    }
15
16    try
17    {
18        $res = $pdo_bdd->query( statement: "SELECT * FROM sondage" );
19    }
20    catch (Exception $e )
21    {
22        die ('Echec lors de l\'exécution de la requête: ('.$e->getMessage() .') ' . $e->getCode());
23    }
24
25    //Traitement de la requête
26    $sHtml='<table>
27
28        <tr><th>Pseudo</th><th>Réponse</th><th>Date</th></tr>';
29
30    $qTab=$res->fetchAll( fetch_style: PDO::FETCH_ASSOC);
31    foreach ($qTab as $tuple) {
32
33        $sHtml.= "<tr><td>".$tuple["nom"]."</td>
34
35        <td>".$tuple["resultat"]."</td>
36
37        <td>".$tuple["date"]."</td>
38
39        </tr>";
40    }
41
42    $sHtml.="</table>";
43
44    ?>
45
46    <!doctype html>
47    <html lang="fr">
48    <head>
49        <title>Réponses</title>
50        <meta charset="utf-8">
51        <meta http-equiv="Content-Style-Type" content="text/css" />
52    </head>
53
54    <body>
55        <?php echo $sHtml ?>
56        <a href="sondage.html">retour vers le sondage</a>
57    </body>
58    </html>

```

Construction du tableau html
qui reflète le contenu de a BDD

Affichage de la base : statistiques.php

```

1 <?php
2 //connexion à la bdd
3 include_once 'config_bd.php';
4 if (isset($_REQUEST['form_sondage'])) { //Ecriture et execution de la requete
5
6     $requete = "INSERT INTO sondage (nom, resultat) VALUES ('".$_REQUEST['pseudo']."' , '".$_REQUEST['choix']."' )";
7     try
8     {
9         $res = $pdo_bdd->query($requete);
10     }
11     catch(Exception $e)
12     {
13         die ('Echec lors de l\'exécution de la requête: ('.$e->getMessage() .') ' . $e->getCode());
14     }
15
16     try
17     {
18         $res = $pdo_bdd->query( statement: "SELECT * FROM sondage" );
19     }
20     catch (Exception $e )
21     {
22         die ('Echec lors de l\'exécution de la requête: ('.$e->getMessage() .') ' . $e->getCode());
23     }
24
25     //Traitement de la requête
26     $sHtml='<table>
27         <tr><th>Pseudo</th><th>Réponse</th><th>Date</th></tr>';
28     $qTab=$res->fetchAll( fetch_style: PDO::FETCH_ASSOC);
29     foreach ($qTab as $tuple) {
30         $sHtml.= "<tr><td>".$tuple["nom"]."</td>
31             <td>".$tuple["resultat"]."</td>
32             <td>".$tuple["date"]."</td>
33             </tr>";
34     }
35     $sHtml.="</table>";
36
37 <?>
38 <!doctype html>
39 <html lang="fr">
40 <head>
41     <title>Réponses</title>
42     <meta charset="utf-8">
43     <meta http-equiv="Content-Style-Type" content="text/css" />
44 </head>
45 <body>
46     <?php echo $sHtml ?>
47     <a href="sondage.html">retour vers le sondage</a>
48 </body>
49 </html>

```

Affiche le html généré

Pseudo	Réponse	Date
brett	1	2007-12-03 09:37:20
Martin	2	2007-12-03 09:37:28
Georges	1	2007-12-03 09:37:40
Anaïs	Acun	2007-12-03 09:37:49
John	Plus de 3	2007-12-03 09:37:59
Luna	2	2007-12-03 09:38:31