

Introduction à PHP

PHP et SGBD : un exemple avec Mysql (MariaDB)

monnerat@u-pec.fr

7 mai 2019

IUT de Fontainebleau

Introduction

Mysql (MariaDB)

API mysqli

Requêtes préparées

Introduction

Sites dynamiques \rightsquigarrow Informations dynamiques

Comment la gérer ?

C(reate), R(ead), U(pdate), D(elete)

Plusieurs solutions :

- Fichiers : difficultés à accéder/organiser l'information logiquement.
Suffisant par exemple pour un compteur de visite !
- Utilisation d'un SGBD : il en existe beaucoup !

ORACLE



mongoDB



Interfaçage de l'application WEB (scripts php) avec le(s) sgbd(s)

2 solutions





- ↪ Utilisation d'une api spécifique au sgbd
api MySQL, api Oracle, api Access, api PostgreSQL, etc.

- ↪ Utilisation d'une api indépendante du sgbd
Une seule interface : c'est le cas avec ODBC (Open DataBase Connectivity), ou encore PDO (PHP Data Objects).

Mysql (MariaDB)

MariaDB est une base de données (disponible sur toutes les plates-formes) implémentant le langage de requête SQL un langage relationnel très connu. Cette partie suppose connue les principes des bases de données relationnelles.

Quelques liens utiles :

- [phpmyadmin](http://www.phpmyadmin.net) : outil libre et gratuit développé par la communauté des programmeurs libres permettant l'administration aisée des bases de données MySQL avec php. Il est disponible sur : <http://www.phpmyadmin.net> .
- Le site <http://www.mysql.com/> 
- La documentation de MySQL est disponibles à <http://www.mysql.com/documentation/> , ainsi qu'en français chez nexen : <http://dev.nexen.net/docs/mysql/> .

Modèle relationnel

- Modèle le plus répandu et le plus classique.
- Les données sont organisées en tables, chacune des colonnes représentant un attribut des données.

Prénom	Nom	Date de naissance
Jean	Dupond	1998-07-12
Marcel	Durand	1951-12-12
Serge	Lang	NULL

- Chaque attribut (colonne) est typé.
- SQL (Structured Query Language) : langage standard de requête et de mise à jour des données (petites variantes suivant les SGBD)

SGBD relationnels les plus connus

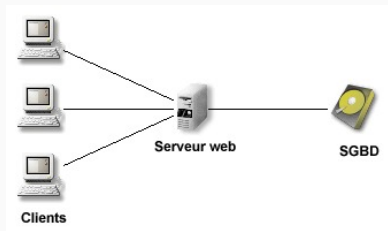
sgbds

Oracle	commercial, le plus utilisé en entreprise
IBM DB2	commercial
Microsoft SQL Server	commercial
Microsoft Access	commercial, pauvre en fonctionnalités
MySQL	libre (?), le plus utilisé sur le Web
PostGreSQL	libre, plus abouti que MySQL
Sqlite	libre, pas besoin de serveur



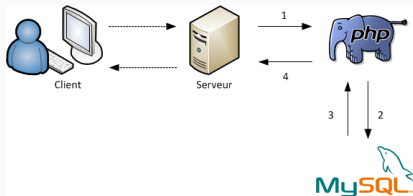
Je vous renvoie à votre cours !

Architecture



Le php intègre une api mysql qui lui permet de faire des requêtes sur un serveur mysql et d'exploiter leurs résultats.

L'application utilise des données gérées par un sgbd.



Architecture Client/serveur, communication via TCP/IP.

- Un serveur (daemon) `mysqld` (port 3306).

```
[denis@portable_denis ~]$ ps -u mysql u
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
mysql    751  0.0  1.1 533740 48468 ?        Ssl  11:16   0:00 /usr/bin/mysqld --pid-file=/run/mysqld/mysqld.pid
[denis@portable_denis ~]$
```

- Une multitude de clients : client en mode console : `mysql`

```
|      220 | Van Cleef      | Lee      |      NULL |
|      221 | Volonte       | Gian Maria |      NULL |
|      224 | Swank         | Hillary  |      1974 |
+-----+-----+-----+-----+
193 rows in set (0.07 sec)

mysql> select * from Artiste LIMIT 0,10;
+-----+-----+-----+-----+
| idArtiste | nom          | prenom   | anneeNaiss |
+-----+-----+-----+-----+
|      1   | Lynch       | David   |      1946 |
|      3   | Hitchcock   | Alfred  |      1899 |
|      4   | Scott       | Ridley  |      1937 |
|      5   | Weaver     | Sigourney |      1949 |
|      6   | Cameron    | James   |      1954 |
|      9   | Tarkovski  | Andrei  |      1932 |
|     10   | Woo        | John    |      1946 |
|     11   | Travolta   | John    |      1954 |
|     12   | Cage       | Nicolas |      1964 |
|     13   | Burton    | Tim     |      1958 |
+-----+-----+-----+-----+
10 rows in set (0.00 sec)

mysql>
```

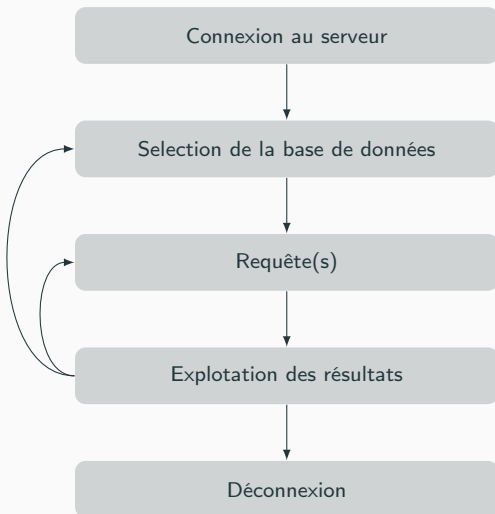
phpmyadmin : interface web (en php) d'administration

The screenshot displays the phpMyAdmin web interface in a browser window. The address bar shows the URL: localhost/phpMyAdmin/index.php?token=d443762b2f5f1a9ce785378b11e83cdf. The interface is organized into several sections:

- Navigation:** A top menu with links for "Bases de données", "SQL", "Etat", "Variables", "Jeux de caractères", "Moteurs", "Privèges", "Log binaire", "Processus", "Exporter", and "Importer".
- Actions:** A section with links for "Modifier le mot de passe" and "Quitter".
- MySQL localhost:** A section for creating a new database, featuring a text input field, a dropdown menu for "Interclassement" (set to "Interclassement"), and a "Créer" button. Below it, another dropdown for "Interclassement pour la connexion MySQL:" is also set to "Interclassement".
- Interface:** A section for user preferences, including a language dropdown (set to "Français - French"), a theme dropdown (set to "Original"), a "Couleur au choix:" button with a "Réinitialiser" option, and a text size dropdown (set to "82%").
- Server Information (MySQL):**
 - Serveur: Localhost via UNIX socket
 - Version du serveur: 5.5.30-log
 - Version du protocole: 10
 - Utilisateur: root@localhost
 - Jeux de caractères pour MySQL: UTF-8 Unicode (utf8)
- Server Information (Serveur web):**
 - Apache/2.2.25 (Unix) mod_ssl/2.2.25 OpenSSL/1.0.1f DAV/2 PHP/5.5.6
 - Version du client MySQL: mysqlnd 5.0.11-dev - 20120503 - \$Id: 40933630edef551dfaca71298a83fad8d03d62d4\$
 - Extension PHP: mysql
- phpMyAdmin:**
 - Version: 3.2.5
 - Documentation
 - Wiki

API mysql

Intéraction Script(s)↔SGBD



L'extension procédurale `mysql` est obsolète depuis PHP 5.5.0, car elle ne prend pas en charge les nouvelles fonctionnalités de `mysql5` : procédures stockées, fonctions, transactions. Pas de requêtes préparées, gestion manuelle des échappements. Elle sera supprimée dans un futur proche.

À la place, soit l'extension `mysqli` ou `PDO_MySQL` devrait être utilisée.

Dans la suite, on présente l'extension **mysqli**. C'est une extension objet, mais qui offre une version procédurale que l'on utilisera.

Dans la plupart des cas, les noms de fonctions ne diffèrent (de celle de mysql) que par leurs préfixes. Quelques fonctions mysqli prennent un gestionnaire de connexion comme premier argument, alors que la fonction correspondante de l'ancienne interface mysql le prenait comme argument optionnel en dernière position.

Un exemple

On dispose d'une base de données `cinema` qui permet de gérer une filmothèque. En particulier, la table `Film` décrite par la relation suivante :

Film (idFilm, titre, annee, #idMes, #genre, resume, #codePays)

permet de stocker le titre et l'année des films de la filmothèque.

But

Écrire une page qui affiche la liste de tous nos titres, avec l'année de sortie.

1. Connexion

```
$link=mysqli_connect("host","user","passwd","cinema");  
if(!$link){  
    die("<p>connexion impossible</p>");  
}
```

2. Requête

```
$resultat=mysqli_query($link,  
    "SELECT titre,annee FROM Film");
```

3. Exploitation

```
if($resultat) {  
    echo "<ul>";  
    while($film=mysqli_fetch_assoc($resultat)){  
        echo "<li>".  
            $film['titre'].  
            " en ".  
            $film['annee'].  
            "</li>";  
    }  
    echo "</ul>";  
} else  
    die("<p>Erreur dans l'exécution de la requête.</p>");
```

4. Déconnexion

```
mysqli_close($link);
```

Fonction de connexion

```
mysqli mysqli_connect (  
    [, string $host = ini_get("mysqli.default_host")  
    [, string $username = ini_get("mysqli.default_user")  
    [, string $passwd = ini_get("mysqli.default_pw")  
    [, string $dbname = ""  
    [, int $port = ini_get("mysqli.default_port")  
    [, string $socket = ini_get("mysqli.default_socket") ]]]]] )
```

Exemple :

```
$link = mysqli_connect(
    'localhost',
    'my_user',
    'my_password',
    'my_db'
);

if (!$link) {
    die('Erreur de connexion (' . mysqli_connect_errno() . ') '
        . mysqli_connect_error());
}

echo 'Succès... ' . mysqli_get_host_info($link) . "\n";

mysqli_close($link);
```

- La fonction

```
mysqli_connect()
```

renvoie un identifiant de connexion, généralement nécessaire par la suite. Renvoie FALSE en cas d'échec.

- La fonction

```
mysqli_select_db()
```

permet de sélectionner une base de données. Renvoie TRUE ou FALSE.

Requêtes

Il existe plusieurs types de requêtes :

- Lecture.
- Modification/Insertion.
- Suppression.

Avec PHP

On utilise directement le langage SQL à partir de php.

La fonction

```
mixed mysqli_query (mysqli $link , string $query )
```

permet d'envoyer une requête au serveur, et retourne le resultat de l'exécution sous forme d'un objet `mysqli_result` ou un booléen.

Caractères spéciaux

Certains caractères dans des paramètres de bases de données mysql doivent être protégés (apostrophe par exemple). Il faut (on peut) utiliser une fonction spécifique à mysql :

```
string mysqli_real_escape_string (  
    mysqli $link ,  
    string $escapestr );
```

```
$nom = mysqli_real_escape_string($1,$_POST['nom']);  
$prenom = mysqli_real_escape_string($1,$_POST['prenom']);  
$adresse = mysqli_real_escape_string($1,$_POST['adresse']);  
$conn = mysqli_connect("localhost","user","pass","db");  
mysqli_query($conn, "INSERT INTO etudiants VALUES  
    ('$nom', '$prenom', '$adresse')"  
);
```


Sélection

Pour une requête de sélection, la fonction `mysqli_query` renvoie un `mysqli_result` qui représente le jeu de résultats sélectionnés (analogie avec les curseurs en PL/SQL). On peut récupérer un à un les résultats avec :

<code>mysqli_fetch_array()</code>	retourne un enregistrement sous la forme d'un tableau associatif, numérique, ou les deux.
<code>mysqli_fetch_object()</code>	retourne un enregistrement sous la forme d'un objet
<code>mysqli_fetch_row()</code>	retourne un enregistrement sous la forme d'un tableau numérique
<code>mysqli_fetch_assoc()</code>	retourne un enregistrement sous la forme d'un tableau associatif

La fonction `mysqli_num_rows()` permet de connaître le nombre d'enregistrements retournés par une requête de sélection.

```
$result=mysqli_query($conn,"SELECT nom,login FROM etudiant");

// version mysql_fetch_array
while ($ligne=mysqli_fetch_assoc($result)){
    echo "<li>Nom : ".$ligne['nom'].
        "login : ".$ligne['login']."</li>";
}

// version mysql_fetch_row
while ($ligne=mysqli_fetch_row($result)){
    echo "<li>Nom : ".$ligne[0].
        "login : ".$ligne[1]."</li>";
}

// version mysql_fetch_object
while ($ligne=mysqli_fetch_object($result)){
    echo "<li>Nom : ".$ligne->nom."login : ".
        $ligne->login."</li>";
}
```

Remarque : depuis php 5.4, il est possible d'utiliser foreach pour itérer un mysqli_result :

```
<?php

$resultat=mysqli_query($link,"SELECT titre,annee
                        FROM Film");

if($resultat) {
    echo "<ul>";
    foreach($resultat as $enr){
        echo "<li>".$enr['titre']." en ".$enr['annee']." </li>";
    }
    echo "</ul>";
}
?>
```

Insertion/Modification

Les requêtes de type INSERT, UPDATE, REPLACE ou DELETE ne retournent normalement pas de données. Néanmoins, il est possible de savoir combien d'enregistrement effectivement insérés, modifiés ou supprimés lors de la **dernière requête** avec la fonction

```
mysqli_affected_rows()
```

```
$query="UPDATE etudiant
      SET age=age+1
      WHERE anniversaire='1995-11-11';";
mysqli_query($conn,$query);
echo mysqli_affected_rows($conn)."etudiants concernes";
```

Dans le cas d'un SELECT `mysqli_affected_rows()` se comporte comme `mysqli_num_rows`.

Remarque : lors d'une requête d'insertion dans une table contenant un champs autoincrémenté, on peut récupérer sa valeur pour la dernière insertion avec

```
mysqli_insert_id()
```

```
<?php
$query="INSERT INTO etudiant ....";
mysqli_query($conn,$query);
echo "Identifiant etudiant : ".mysqli_insert_id($conn);
?>
```

La connexion au serveur est fermée automatiquement à la fin du script PHP. On peut néanmoins le faire manuellement à l'aide la fonction

```
mysqli_close()
```

Utiliser la fonction

- `mysqli_real_escape_string`

pour protéger les caractères apostrophes, antislashes et guillemets pour mysql.

- `trim`

qui supprime les espaces en début et fin de mot, avant insertion dans la base.

- `htmlentities`

qui permet de convertir les caractères spéciaux d'HTML : par exemple `&`, `<`, `>`, `"`, `'`.

- `nl2br`

de manière à convertir les sauts lignes en balise `
`.

Valider et nettoyer les données "externes" avec les fonctions `filter_var` et/ou `filter_input`.

Vue générale de l'api mysql

<code>mysqli_affected_row()</code>	nombre d'enregistrements concernés par la dernière requête.
<code>mysqli_close()</code>	Deconnexion.
<code>mysqli_connect()</code>	Connexion à un serveur.
<code>mysqli_data_seek()</code>	déplace le pointeur interne de résultat.
<code>mysqli_query()</code>	Exécution d'une requête.
<code>mysqli_error()</code>	Retourne le texte associé à l'erreur provoqué par la dernière requête.
<code>mysqli_escape_string()</code>	Protège une chaîne pour la passer à <code>mysql_query</code> .
<code>mysqli_fetch_array()</code>	Lit une ligne de résultat dans tableau.
<code>mysqli_fetch_assoc()</code>	Idem, dans un tableau associatif.
<code>mysqli_fetch_object()</code>	Idem, dans un objet.
<code>mysqli_fetch_row()</code>	Idem, dans un tableau numérique.

api suite

<code>mysqli_free_result()</code>	Efface le résultat de la mémoire.
<code>mysqli_insert_id()</code>	Retourne l'identifiant de la dernière requête INSERT.
<code>mysqli_num_fields()</code>	Retourne le nombre de champs d'un résultat.
<code>mysqli_fetch_fields</code>	Retourne le prochain champs dans le jeu de résultats.
<code>mysqli_num_rows()</code>	Retourne le nombre de lignes d'un resultat de SELECT.
<code>mysqli_select_db()</code>	Selectionne une base de données.

Requêtes préparées

La base de données MySQL supporte les requêtes préparées. Une requête préparée ou requête paramétrable est utilisée pour exécuter la même requête plusieurs fois, avec une grande efficacité.

Deux étapes : la préparation et l'exécution.

Lors de la préparation, un template de requête est envoyé au serveur de base de données. Le serveur effectue une vérification de la syntaxe, et initialise les ressources internes du serveur pour une utilisation ultérieure.

Le serveur MySQL supporte le mode anonyme, avec des marqueurs de position utilisant le caractère ?.

```
mysqli_stmt mysqli_prepare (mysqli $link , string $query)
```

Exemple

```
$link = mysqli_connect("localhost","","","cinema");  
if (!$link) die ("pb");  
$stmt = mysqli_prepare(  
    $link,  
    "INSERT  
    INTO Artiste (nom,prenom,anneeNaiss)  
    VALUES (?,?)"  
);  
if (!$stmt) die ("pb");
```

Notez bien la présence des marqueurs "?".

Liaison

```
bool mysqli_stmt_bind_param (
    mysqli_stmt $stmt ,
    string $types ,
    mixed &$var1 [, mixed &$var2, ... ] )
```

types est une chaîne de caractères qui spécifie les types des variables à lier :

Caractère	Description
i	correspond à une variable de type entier
d	correspond à une variable de type nombre décimal
s	correspond à une variable de type chaîne de caractères
b	correspond à une variable de type BLOB, qui sera envoyé par paquets

Exécution

```
bool mysqli_execute ( mysqli_stmt $stmt )
```

Exemple

```
$nom = "Ford";  
$prenom = "John";  
$annee = 1894;  
mysqli_stmt_bind_param($stmt, "ssi", $nom, $prenom, $annee);  
mysqli_execute($stmt);
```

Si la requête est UPDATE, DELETE ou INSERT, le nombre total de lignes affectées est disponible via la fonction

```
int mysqli_stmt_affected_rows ( mysqli_stmt $stmt )
```

Si la fonction génère un résultat, il sera disponible via la fonction

```
bool mysqli_stmt_fetch ( mysqli_stmt $stmt )
```

après avoir lié le résultat avec des variables par

```
bool mysqli_stmt_bind_result (
    mysqli_stmt $stmt ,
    mixed &$var1 [, mixed &$var2, ] )
```


Récupérer les résultats un à un

```
$stmt = mysqli_prepare(
    $link,
    "SELECT nom,prenom,anneeNaiss
    FROM Artiste
    WHERE anneeNaiss > 1960");
if (!$stmt) die("pb");
mysqli_execute($stmt);
mysqli_stmt_bind_result($stmt,$nom,$prenom,$annee);
while(mysqli_stmt_fetch($stmt)){
    echo "$nom $prenom : $annee";
}
```

Compter le nombre de résultats

```
mysqli_stmt_store_result($stmt);
$num = mysqli_stmt_num_rows($stmt);
```

On peut aussi récupérer le résultat avec

```
mysqli_result mysqli_stmt_get_result ( mysqli_stmt $stmt )
```

que l'on peut traiter de façon usuelle :

```
$stmt = mysqli_prepare(
    $link,
    "SELECT nom,prenom,anneeNaiss
    FROM Artiste
    WHERE anneeNaiss > 1960");
if (!$stmt) die("pb");
mysqli_execute($stmt);

$res = mysqli_stmt_get_result($stmt);
$nb = mysqli_num_rows($res);
while ($enr = mysqli_fetch_assoc($res)){
    echo $enr['nom']." ".$enr['prenom'];
}
```