

Se connecter à une base de données MySQL en PHP

Télécharger le PDF du cours Dans ce nouveau chapitre, nous allons passer en revue les différents moyens que nous avons de nous connecter au serveur et donc à nos bases de données MySQL en PHP.

Nous discuterons des avantages et des inconvénients de telle ou telle méthode et allons également apprendre à nous connecter à nos bases de données à proprement parler.



Se connecter à MySQL en PHP : les API proposées par le PHP

Pour pouvoir manipuler nos bases de données MySQL en PHP (sans passer par phpMyAdmin), nous allons déjà devoir nous connecter à MySQL.

Pour cela, le PHP met à notre disposition deux API (Application Programming Interface):

- L'extension MySQLi;
- L'extension PDO (PHP Data Objects).

Note: Auparavant, nous pouvions également utiliser l'extension MySQL. Cependant, cette extension est désormais dépréciée et a été remplacée par MySQLi (« i » signifie « improved », c'est-à-dire « amélioré » en français).

Cours complet PHP et MySQL

PIERRE GIRAUD

INTRODUCTION AU COURS PHP ET MYSQL

- 1. Introduction au cours : définitions et rôles du PHP et du MySQL
- 2. Client et serveur : définitions et interactions
- 3. Mise en place de notre environnement de travail
- 4. Créer, enregistrer et exécuter un script PHP
- 5. Afficher un sultat en PHP avec

Quelle API préférer : MySQLi ou PDO ?

Le PHP nous fournit donc deux API pour nous connecter à MySQL et manipuler nos bases de données.

Chacune de ces deux API possède des forces différentes et comme vous vous en doutez elles ne sont pas forcément interchangeables.

Il existe notamment une différence notable entre ces deux API : l'extension MySQLi ne va fonctionner qu'avec les bases de données MySQL tandis que PDO va fonctionner avec 12 systèmes de bases de données différents.

Pour cette raison, nous préférerons généralement le PDO car si vous devez un jour utiliser un autre système de bases de données, le changement sera beaucoup plus simple que si vous avez tout codé en MySQLi auquel cas vous devrez réécrire le code dans son ensemble.

ement sites web 1€/mois - 100% illimité, e Gratuit



- 7. Les types de données en PHP
- 8. Opérateurs et concaténation en PHP

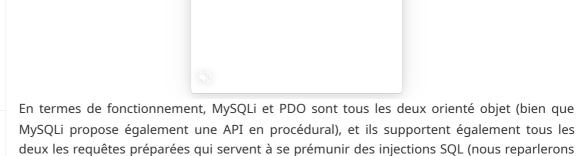
LES STRUCTURES DE CONTRÔLE EN PHP

- 9. Présentation des conditions et des opérateurs de comparaison
- 10. Les conditions if, if...else et if...elseif... else
- 11. Créer des conditions robustes avec les opérateurs logiques
- 12. Ecrire des conditions condensées avec les opérateurs ternaire et fusion null
- 13. L'instruction switch en PHP
- 14. Les boucles PHP et les opérateurs d'incrémentation et de décrémentation
- 15. Inclure des fichiers dans un autre en PHP avec include et require

DÉCOUVERTE DES FONCTIONS EN PHP

16. Introduction aux fonctions PHP

¥ 17. Contrôler I



Dans ce cours, j'utiliserai donc PDO sauf pour ce chapitre où il me semble intéressant de vous montrer les différences d'écriture pour un script de connexion à une base de données MySQL.

Connexion au serveur avec MySQLi orienté objet

de cela dans la suite du cours).

Pour se connecter au serveur et accéder à nos bases de données MySQL en MySQLi orienté objet, nous allons avoir besoin de trois choses : le nom du serveur ainsi qu'un nom d'utilisateur (avec des privilèges de type administrateur) et son mot de passe.

Dans le cas où votre site est hébergé sur un serveur, votre hébergeur vous donnera ces différents éléments. Ici, bien évidemment, nous travaillons en local. Le nom de notre serveur est donc localhost.

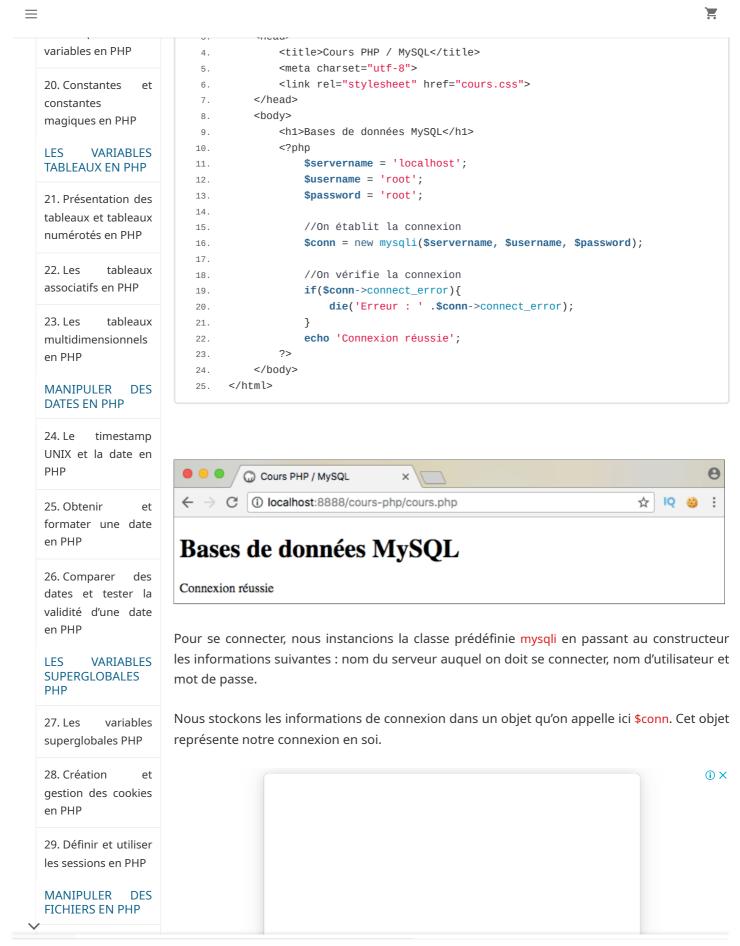


Concernant les identifiants au serveur local, ils peuvent changer selon vos paramétrages et selon le système que vous utilisez. Cependant, si vous disposez des réglages par défaut, le nom d'utilisateur devrait toujours être root et le mot de passe associé devrait être soit root soit une chaine de caractère vide.

ement sites web 1€/mois - 100% illimité, e Gratuit

le données, SSL, Bande passante illimitée, Baie de stockage SSD hosteur.com ites

(i) X



ement sites web 1€/mois - 100% illimité, e Gratuit

PHP

Ħ

32. Créer et écrire dans un fichier en PHP

33. Autres opérations sur les fichiers en PHP

UTILISER LES EXPRESSIONS RÉGULIÈRES OU RATIONNELLES EN PHP

34. Introduction aux expressions rationnelles ou expressions régulières

35. Les fonctions PCRE PHP

36. Les classes de caractères des regex

37. Les métacaractères des regex PHP

38. Les options des expressions régulières disponibles en PHP

PROGRAMMATION ORIENTÉE OBJET (POO) PHP : CONCEPTS DE BASE

39. Introduction à la programmation orientée objet PHP : classes, instances et objets

40. Propriétés et méthodes en PHP orienté objet

11 Inc. máthadac

informations relatives a la tentative de connexion.

Pour cela, nous utilisons la propriété connect_error de la classe mysqli qui retourne un message d'erreur relatif à l'erreur rencontrée en cas d'erreur de connexion MySQL ainsi que la fonction die() pour stopper l'exécution du script en cas d'erreur.

Attention : La propriété connect_error de mysqli ne fonctionne correctement que depuis la version 5.3 de PHP. Utilisez la fonction mysqli_connect_error() pour les versions antérieures. Notez ici qu'on aurait également pu utiliser les exceptions et des blocs try et catch pour gérer les erreurs potentielles. Je voulais juste vous présenter une autre manière de faire ici.

Dans le cas où la connexion réussit, on se contente d'afficher un message « connexion réussie ».

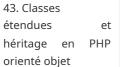
Si vous désirez la liste complète des propriétés et méthodes de la classe mysqli, je vous invite à consulter la <u>documentation officielle</u>.

Connexion au serveur avec MySQLi procédural

Nous allons également pouvoir utiliser un script en procédural avec MySQLi pour nous connecter au serveur et à la base de données MySQL.

```
<!DOCTYPE html>
 1.
     <html>
 2.
 3.
          <head>
              <title>Cours PHP / MySQL</title>
 4.
              <meta charset="utf-8">
              <link rel="stylesheet" href="cours.css">
 7.
          <body>
 8.
9.
              <h1>Bases de données MySQL</h1>
              <?php
                  $servername = 'localhost';
11.
                  $username = 'root';
12.
                  $password = 'root';
13.
                  //On établit la connexion
                  $conn = mysqli_connect($servername, $username, $password);
16.
17.
                  //On vérifie la connexion
                  if(!$conn){
                      die('Erreur : ' .mysqli_connect_error());
20.
                  }
21.
22.
                  echo 'Connexion réussie';
              ?>
          </body>
24.
     </html>
25.
```

ement sites web 1€/mois - 100% illimité, e Gratuit



- 44. Surcharge
 d'éléments et
 opérateur de
 résolution de portée
 en PHP
- 45. Les constantes de classe en PHP objet
- 46. Les propriétés et méthodes statiques en PHP objet
- 47. Les méthodes et les classes abstraites en PHP objet
- 48. Les interfaces en PHP orienté objet
- 49. Les méthodes magiques en orienté objet PHP

PROGRAMMATION ORIENTÉE OBJET PHP : NOTIONS AVANCÉES

- 50. Le chainage de méthodes en PHP
- 51. Les closures et les classes anonymes en PHP objet
- 52. L'auto
 chargement des
 classes en PHP
- 53. Le mot clef final en PHP objet

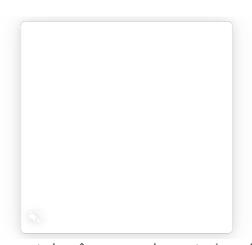
✓ 54. La résolution

Bases de données MySQL

Connexion réussie

Ce script ressemble à priori au précédent et pourtant il est bien très différent : nous n'avons cette fois-ci plus recours à notre classe mysqli ni à l'orienté objet.

A la place, nous utilisons les fonctions mysqli_connect() pour nous connecter à la base de données et mysqli_connect_error() pour obtenir des informations sur l'erreur de connexion si il y en a une.



En dehors de ça, le principe reste le même : nous devons toujours fournir le nom du serveur ainsi que des identifiants de connexion (nom d'utilisateur et mot de passe) pour se connecter avec la fonction mysqli_connect() et nous prenons toujours en charge les cas d'erreur de connexion et stoppant l'exécution du script avec la fonction die().

Connexion au serveur avec PDO

Pour se connecter en utilisant PDO, nous allons devoir instancier la classe PDO en passant au constructeur la source de la base de données (serveur + nom de la base de données) ainsi qu'un nom d'utilisateur et un mot de passe.

ement sites web 1€/mois - 100% illimité, e Gratuit

le données, SSL, Bande passante illimitée, Baie de stockage SSD hosteur.com ites

Ħ

 \bigcirc X





56. L'interface Iterator et le parcours d'objets en PHP

57. Le passage d'objets en PHP : identifiants et références

58. Le clonage d'objets et la méthode magique PHP__clone()

59. La comparaison d'objets PHP

ESPACES DE NOMS, FILTRES ET GESTION DES ERREURS EN PHP

60. Les espaces de noms PHP

61. Présentation des filtres PHP

62. Filtres de validation, de nettoyage et drapeaux de l'extension PHP Filter

63. Utilisation pratique des filtres en PHP

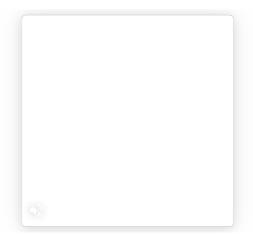
64. Définition et gestion des erreurs en PHP

65. Déclenchement, capture et gestion des exceptions PHP: try, throw, catch

INTRODUCTION AUX BASES DE DONNÉES. AU SOL Vous pouvez déjà remarquer ici que pour se connecter à une base de données avec PDO, vous devez passer son nom dans le constructeur de la classe PDO. Cela implique donc qu'il faut que la base ait déjà été créée au préalable (avec phpMyAdmin par exemple) ou qu'on la crée dans le même script.

Notez également qu'avec PDO il est véritablement indispensable que votre script gère et capture les exceptions (erreurs) qui peuvent survenir durant la connexion à la base de données.

En effet, si votre script ne capture pas ces exceptions, l'action par défaut du moteur Zend (plus de détail sur le moteur <u>ici</u>) va être de terminer le script et d'afficher une trace. Cette trace contient tous les détails de connexion à la base de données (nom d'utilisateur, mot de passe, etc.). Nous devons donc la capturer pour éviter que des utilisateurs malveillants tentent de la lire.



Pour faire cela, nous allons utiliser des blocs try et catch.

```
1.
     <!DOCTYPE html>
     <html>
 2.
          <head>
 3.
              <title>Cours PHP / MySQL</title>
 4.
              <meta charset="utf-8">
              <link rel="stylesheet" href="cours.css">
 7.
          </head>
          <body>
8.
              <h1>Bases de données MySQL</h1>
9.
              <?php
10.
                  $servername = 'localhost';
11.
                  $username = 'root';
12.
                  $password = 'root';
13.
14.
                  //On essaie de se connecter
```

ement sites web 1€/mois - 100% illimité, e Gratuit

le données, SSL, Bande passante illimitée, Baie de stockage SSD hosteur.com ites

Ħ

(i) X

Ħ

```
MySQL et découverte de PHPMyAdmin
```

68. Se connecter à une base de données MySQL en PHP

69. Créer une base de données MySQL et une table dans la base

MANIPULER DES DONNÉES DANS DES BASES MYSQL AVEC PDO

70. Insérer des données dans une table MySQL

71. Les requêtes MySQL préparées avec PDO PHP

72. Modifier les données d'une table MySQL ou sa structure

73. Supprimer des données, une table ou une base de données MySQL

74. Sélection simple de données dans une table MySQL en PHP

75. Utiliser des critères de sélection pour sélectionner des données dans une table MySQL

76. Utiliser les fonctions d'agrégation et les fonctions scalaires

```
/*On capture les exceptions si une exception est lancée et on
23.
    affiche
                   *les informations relatives à celle-ci*/
24.
                  catch(PD0Exception $e){
25.
                    echo "Erreur : " . $e->getMessage();
26.
27.
              2>
28.
          </body>
29.
30.
     </html>
```

Ici, nous utilisons également la méthode setAttribute() en lui passant deux arguments PDO::ATTR_ERRMODE et PDO::ERRMODE_EXCEPTION.

La méthode setAttribute() sert à configurer un attribut PDO. Dans ce cas précis, nous lui demandons de configurer l'attribut PDO::ATTR_ERRMODE qui sert à créer un rapport d'erreur et nous précisons que l'on souhaite qu'il émette une exception avec PDO::ERRMODE_EXCEPTION.

Plus précisément, en utilisant PDO::ERRMODE_EXCEPTION on demande au PHP de lancer une exception issue de la classe PDOException (classes étendue de Exception) et d'en définir les propriétés afin de représenter le code d'erreur et les informations complémentaires.

Ensuite, nous n'avons plus qu'à capturer cette exception PDOException et à afficher le message d'erreur correspondant. C'est le rôle de notre bloc catch.

ement sites web 1€/mois - 100% illimité, e Gratuit



78. Création de jointures SQL	Pour fermer la connexion avant cela, nous allons devoir utiliser différentes méthodes selon la méthode d'ouverture choisie.
79. L'opérateur SQL UNION	Si on utilise MySQLi orienté objet, alors il faudra utiliser la méthode close()
80. Les opérateurs de sous requête SQL	
GESTION DES FORMULAIRES HTML AVEC PHP	
81. Rappels sur les formulaires HTML	
82. Récupérer et manipuler les données des formulaires HTML en PHP	
83. Sécurisation et validation des formulaires en PHP	
CONCLUSION DU COURS PHP ET MYSQL	
84. Conclusion du cours complet PHP et MySQL	Si on utilise MySQLi procédural, on utilisera la fonction mysqli_close()

ement sites web 1€/mois - 100% illimité, e Gratuit

=		
	Si on utilise PDO, il faudra détruire l'objet représentant la connexion et effacer toutes s	A S
	références. Nous pouvons faire cela en assignant la valeur NULL à la variable gérant l'objet.	
~		

ement sites web 1€/mois - 100% illimité, e Gratuit



Vous devez vous connecter pour publier un commentaire.

Connexion

Confidentialité CGV

/ Sitemap

© Pierre Giraud - Toute reproduction interdite - Mentions légales

ement sites web 1€/mois - 100% illimité,

e Gratuit

le données, SSL, Bande passante illimitée, Baie de stockage SSD hosteur.com

 Ξ