

Connectar base de dades

La connexió a bases de dades MySQL a PHP s'estableix principalment mitjançant l'extensió **MySQLi** (MySQL Improved), que ofereix mètodes orientats a objectes i procedurals. **PDO** (PHP Data Objects) és l'altra alternativa crucial.

Connexió a MySQL amb MySQLi

L'extensió **MySQLi** és la manera recomanada de treballar amb bases de dades MySQL a PHP, ja que és una versió millorada i més ràpida que l'antiga (i obsoleta) extensió `mysql`.

1. Orientat a Objectes (OO)

Aquest és l'estil més modern i preferit.

Funció/Mètode	Paràmetres	Propòsit
<code>new mysqli(...)</code>	<code>\$servername</code> , <code>\$username</code> , <code>\$password</code> , (opcional: <code>\$dbname</code> , <code>\$port</code> , <code>\$socket</code>)	Crea una nova instància de l'objecte <code>mysqli</code> , intentant connectar immediatament.
<code>\$conn->connect_error</code>	(Propietat)	Retorna un missatge d'error de la connexió si falla.
<code>die(...)</code>	(Funció)	Atura l'execució de l'script i mostra un missatge. S'utilitza en cas d'error fatal.

Codi Base (OO):

PHP

```
$conn = new mysqli($servername, $username, $password);  
if ($conn->connect_error) {  
    die("Connection failed: " . $conn->connect_error);  
}
```

2. Procedural

Aquest estil utilitza funcions que comencen amb el prefix `mysqli_`.

Funció	Paràmetres	Propòsit
<code>mysqli_connect(...)</code>	<code>\$servername</code> , <code>\$username</code> , <code>\$password</code> , (opcional: <code>\$dbname</code> , <code>\$port</code> , <code>\$socket</code>)	Intenta connectar i retorna l'enllaç de connexió (<code>\$conn</code>) en cas d'èxit, o <code>FALSE</code> en cas d'error.
<code>mysqli_connect_error()</code>	(Sense paràmetres)	Retorna un missatge d'error de la darrera funció <code>mysqli_connect()</code> que ha fallat.

Codi Base (Procedural):

PHP

```
$conn = mysqli_connect($servername, $username, $password);
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}
```

Errors Més Comuns

Els errors en la connexió solen ser d'aquests tipus:

Error Comú	Causa	Solució Típica
" Access denied for user... "	Credencials incorrectes (usuari o contrasenya).	Verificar la configuració a l'allotjament o al servidor local.
" Unknown database... "	La base de dades especificada no existeix.	Assegurar-se que <code>\$dbname</code> (si s'ha afegit al constructor) sigui correcte.
" Connection refused "	El servidor MySQL no s'està executant o el firewall el bloqueja.	Iniciar el servidor MySQL (p. ex., XAMPP/WAMP/MAMP) o comprovar ports.
" Undefined property... "	Si s'intenta accedir a l'objecte <code>\$conn</code> després que hagi fallat.	La verificació de l'error hauria d'aturar l'script (<code>die()</code>).

Paràmetres Essencials

Els paràmetres bàsics per a la connexió són:

- `$servername`: Normalment `"localhost"` per a entorns de desenvolupament o el nom de l'amfitrió del servidor de BBDD a l'allotjament.
- `$username`: El nom d'usuari de MySQL (p. ex., `"root"` en desenvolupament).
- `$password`: La contrasenya de l'usuari de MySQL.
- **Recomanat:** Afegir el nom de la base de dades (`$dbname`) com a quart paràmetre, que permet seleccionar la BBDD alhora de connectar.

I com funciona amb PDO?

És una excel·lent pràctica! **PDO (PHP Data Objects)** és la manera més recomanada d'establir connexions a bases de dades amb PHP per la seva seguretat i versatilitat. A diferència de MySQLi, PDO és purament **Orientat a Objectes** i s'utilitza una capa d'abstracció (el DSN) per especificar el tipus de base de dades, fent-lo molt més flexible.

Amb PDO, la connexió s'estableix creant una nova instància de l'objecte `PDO`. El procés es realitza dins d'un bloc `try...catch` per gestionar els errors mitjançant **excepcions**, una pràctica de gestió d'errors superior a la funció `die()`.

Codi Base (Orientat a Objectes)

PHP

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "database_name";

// 1. Construcció de la cadena DSN (Data Source Name)
// Especifica el tipus de BDD (mysql) i els detalls de la connexió.
$dsn = "mysql:host=$servername;dbname=$dbname;charset=utf8mb4";

// 2. Opcions de configuració (Recomanades per a seguretat i errors)
$options = [
    // Llança una excepció en cas d'error de SQL (millor gestió)
    PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION,
    // Desactiva l'emulació de sentències preparades (més seguretat contra Injecció
    SQL)
    PDO::ATTR_EMULATE_PREPARES => false,
    // Mode per defecte de retorn de resultats (objectes)
    PDO::ATTR_DEFAULT_FETCH_MODE => PDO::FETCH_ASSOC,
];

try {
    // Crear la connexió
    $pdo = new PDO($dsn, $username, $password, $options);
    echo "Connected successfully (via PDO)";

    // Aquí s'executarien les consultes utilitzant $pdo
} catch (PDOException $e) {
    // Captura els errors de connexió
    // Es mostra el missatge d'error i el codi (codi d'error més baix de la BDD).
    die("Connection failed: " . $e->getMessage() . " (Code: " . $e->getCode() . ")");
}
```

```
// La connexió es tanca automàticament quan l'script finalitza o si es fa $pdo = null;
?>
```

Components de PDO

Component/Funció	Paràmetres/Valor	Propòsit
<code>\$dsn</code> (Data Source Name)	<code>"mysql:host=...;dbname=...;charset=..."</code>	Defineix l'estructura de la connexió (tipus de BDD, servidor, nom de la BDD i codificació). El <code>charset=utf8mb4</code> és crucial per al suport complet d'emojis.
<code>new PDO(...)</code>	<code>\$dsn</code> , <code>\$username</code> , <code>\$password</code> , <code>\$opcions</code>	Crea l'objecte de connexió i inicia l'intent de connexió.
<code>try...catch</code>	<code>PDOException \$e</code>	Mecanisme de gestió d'errors estàndard i superior. Si la connexió falla, el <code>catch</code> intercepta l'error.
<code>\$e->getMessage()</code>	(Mètode)	Obté el missatge d'error de la base de dades (p. ex., "Access denied").
<code>PDO::ATTR_ERRMODE</code>	<code>PDO::ERRMODE_EXCEPTION</code>	Mecanisme Clau: Força PDO a llançar objectes d'excepció en cas d'errors de SQL. Això és molt més net i segur que comprovar errors manuals.
<code>PDO::ATTR_EMULATE_PREPARES</code>	<code>false</code>	Mecanisme Clau de Seguretat: Desactiva l'emulació de sentències preparades de PHP i força el driver a enviar les sentències preparades de manera nativa al servidor de BDD, oferint la màxima protecció contra la Injecció SQL .

Errors en PDO

- DSN Incorrecte:** Un error de sintaxi al `$dsn` (p. ex., oblidar el punt i coma o escriure malament `mysql:host`).
- Credencials:** Usuari/contrasenya incorrectes, com en MySQLi.
- Encoding:** No especificar `charset=utf8mb4` pot provocar problemes amb caràcters especials.

Avantatges PDO

- Seguretat Superior:** Fàcilment configurable per deshabilitar l'emulació de preparació i forçar les sentències preparades segures.
- Uniformitat:** Utilitza el mateix codi per a qualsevol base de dades (MySQL, SQLite, etc.).

- **Gestió d'Errors Moderna:** Utilitza excepcions, que permeten al codi continuar o gestionar l'error de manera més elegant que amb un `die()`.

El format utilitzat en les cadenes de connexió **DSN (Data Source Name)** serveix per proporcionar a l'extensió **PDO** de PHP tota la informació necessària per connectar-se a una base de dades (BDD) específica, independentment del seu tipus (MySQL, PostgreSQL, SQLite, etc.).

Estructura del DSN

El format DSN és una cadena de text amb una sintaxi molt específica. La cadena es divideix en dues parts principals separades pel signe de dos punts (:):

Driver (Tipus BBDD)

La primera part especifica el controlador de la base de dades que PDO ha d'utilitzar. Aquesta és la clau de la flexibilitat de PDO.

- **Exemples de Drivers:**

- `mysql`
- `pgsql` (PostgreSQL)
- `sqlite`
- `sqlsrv` (Microsoft SQL Server)

Paràmetres específics

La segona part (després dels dos punts) conté els detalls de la connexió. Aquests paràmetres s'especifiquen com a parelles `nom=valor` separades per un punt i coma (;). Els paràmetres més comuns per a una connexió **MySQL** són:

- **host**: El nom o adreça IP del servidor de la BBDD (p. ex., `localhost`).
- **dbname**: El nom de la base de dades a la qual es vol connectar.
- **charset**: La codificació de caràcters a utilitzar (p. ex., `utf8mb4` per al suport complet d'emojis i caràcters especials).

Exemple MySQL

La cadena DSN que has vist als exemples de PDO es construeix així:

PHP

```
$servername = "localhost";
$dbname = "base_de_dades_prova";

$dsn = "mysql:host=$servername;dbname=$dbname;charset=utf8mb4";
//      ^^^^^ ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
//      Driver      Host + Valor      BDD + Valor      Charset + Valor
```

Aquesta cadena `$dsn` és un dels tres paràmetres essencials que es passen al constructor de l'objecte PDO, juntament amb el nom d'usuari i la contrasenya:

PHP

```
$pdo = new PDO($dsn, $username, $password);
```

↻ Revisió núm. 3

★ Admin l'ha creat 2025-10-14 10:40:57 UTC

✎ Admin l'ha actualitzat 2025-10-14 16:35:45 UTC