



# Connecting to MySQL

**Summary:** in this tutorial, you'll learn step by step how to connect to a MySQL database from PHP using PDO.

## Prerequisites

Before connecting to a MySQL database server, you need to have:

- A MySQL database server, a database, and an account that has access to the database.
- PDO MySQL driver enabled in the php.ini file

## 1) Setting MySQL database parameters

Suppose you have a local MySQL database server that has the following information:

- The host is `localhost`.
- The `bookdb` database on the local database server.
- The account with the user `root` and password `'S@cr@t1!'` that can access the `bookdb` database.

In PHP, you can create a `config.php` file and place the database parameters:

```
<?php
```

```
$host = 'localhost';  
$db = 'bookdb';  
$user = 'root';  
$password = 'S@cr@t1!';
```

To use the database parameters, you can include the `config.php` file using the `require` (<https://www.phptutorial.net/php-tutorial/php-require/>) construct:

```
<?php
```

```
require 'config.php';
```

## 2) Enable PDO\_MySQL Driver

PDO\_MYSQL is a driver that implements the PDO interface. PDO uses the PDO\_MYSQL driver to connect to a MySQL database.

To check if the PDO\_MYSQL driver is enabled, you open the `php.ini` file. The `php.ini` file is often located under the `php` directory. For example, you can find the `php.ini` file under the `C:\xampp\php` directory if you use `XAMPP` on Windows.

The following shows the extension line in the `php.ini` file:

```
;extension=php_pdo_mysql.dll
```

To enable the extension, you need to uncomment it by removing the semicolon ( `;` ) from the beginning of the line like this:

```
extension=php_pdo_mysql.dll
```

After that, you need to restart the web server for the change to take effect.

## MySQL data source name

PDO uses a data source name (DSN) that contains the following information:

- The database server host
- The database name
- The user
- The password
- and other parameters such as character sets, etc.

PDO uses this information to make a connection to the database server. To connect to the MySQL database server, you use the following data source name format:

```
"mysql:host=host_name;dbname=db_name;charset=UTF8"
```

For example:

```
$dsn = "mysql:host=localhost;dbname=bookdb;charset=UTF8";
```

Note that the charset UTF-8 sets the character set of the database connection to UTF-8.

## Connecting to MySQL

The following `index.php` script illustrates how to connect to the `bookdb` database on the MySQL database server with the `root` account:

```
<?php

require 'config.php';

$dsn = "mysql:host=$host;dbname=$db;charset=UTF8";

try {
    $pdo = new PDO($dsn, $user, $password);

    if ($pdo) {
        echo "Connected to the $db database successfully!";
    }
} catch (PDOException $e) {
    echo $e->getMessage();
}
```

How it works.

- First, create a new PDO object with the data source name, user, and password. The PDO object is an instance of the `PDO` class.

- Second, show the success message if the connection is established successfully or an error message if an error occurs.

If you have everything set up correctly, you will see the following message:

```
Connected to the bookdb database successfully!
```

## Error handling strategies

PDO supports three different error handling strategies:

- `PDO::ERROR_SILENT` – PDO sets an error code for inspecting using the `PDO::errorCode()` and `PDO::errorInfo()` methods. The `PDO::ERROR_SILENT` is the default mode.
- `PDO::ERRMODE_WARNING` – Besides setting the error code, PDO will issue an `E_WARNING` message.
- `PDO::ERRMODE_EXCEPTION` – Besides setting the error code, PDO will raise a `PDOException`.

To set the error handling strategy, you can pass an associative array to the PDO constructor like this:

```
$pdo = new PDO($dsn, $user, $password, [PDO::ATTR_ERRMODE => PDO::ERRMOI
```

---

Or you can use the `setAttribute()` method of the PDO instance:

```
$pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
```

## Troubleshooting

There are some common issues when you connect to a MySQL database:

If the MySQL driver is not enabled in the `php.ini` file, you will get the error message:

```
could not find driver
```

If you provide an incorrect password, you get the following error message:

```
SQLSTATE[HY000] [1045] Access denied for user 'root'@'localhost' (using
```

---

If you provide an invalid database name or the database does not exist, you get the following error message:

```
SQLSTATE[HY000] [1049] Unknown database 'bookdb'
```

If you provide an invalid database hostname, the following error message will display:

```
SQLSTATE[HY000] [2002] php_network_getaddresses: getaddrinfo failed: No such ho
```

---

## Summary

- Enable the `PDO_MYSQL` driver in the `php.ini` file for connecting to a MySQL database from PHP PDO.
- Create an instance of the PDO class to make a connection to a MySQL database.
- Use the PDO constructor or the `setAttribute()` method to set an error handling strategy.