



**Kostenloses eBook**

**LERNEN**

**pdo**

Free unaffiliated eBook created from  
**Stack Overflow contributors.**

**#pdo**

# Inhaltsverzeichnis

Über.....	1
<b>Kapitel 1: Erste Schritte mit pdo.....</b>	<b>2</b>
Bemerkungen.....	2
Examples.....	2
Installation oder Setup.....	2
<b>Zusammenfassung.....</b>	<b>2</b>
<b>Verbindung zur Datenbank mit PDO:.....</b>	<b>2</b>
<b>Sichere Auswahl aus der Datenbank mit PDO:.....</b>	<b>3</b>
Tutorials:.....	3
PDO-Einstellungsattribut-Fehlercode.....	3
<b>Kapitel 2: Bereiten Sie Ihre SQL-Anweisungen vor und führen Sie sie aus.....</b>	<b>7</b>
Bemerkungen.....	7
Examples.....	7
Verwendungszweck.....	7
<b>Credits.....</b>	<b>9</b>



You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [pdo](#)

It is an unofficial and free pdo ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official pdo.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to [info@zzzprojects.com](mailto:info@zzzprojects.com)

---

# Kapitel 1: Erste Schritte mit pdo

## Bemerkungen

Dieser Abschnitt bietet einen Überblick über das, was pdo ist und warum ein Entwickler es verwenden möchte.

Es sollte auch alle großen Themen in pdo erwähnen und auf die verwandten Themen verweisen. Da die Dokumentation für pdo neu ist, müssen Sie möglicherweise erste Versionen dieser verwandten Themen erstellen.

## Examples

### Installation oder Setup

PDO ist eine universelle Datenbank - Verbindung Befehl in PHP, es 12 verschiedene Datenbanktyp zB *MySQL unterstützen, MongoDB, NoSQL*. Ein großer Vorteil von PDO ist, dass der Code zur Unterstützung des Datenbanktyps berechnet wird, sodass Sie beim Wechseln zu einem anderen Datenbanksystem keine Chance haben.

---

## Zusammenfassung

	PDO	MySQLi
<b>Datenbank-Unterstützung</b>	12 verschiedene Fahrer	MySQLi
<b>API</b>	OOP	OOP + verfahrenstechnisch
<b>Verbindung</b>	Einfach	Einfach
<b>Benannte Parameter</b>	Ja	Nein
<b>Objektzuordnung</b>	Ja	Ja
<b>Vorbereitete Anweisungen (Client-Seite)</b>	Ja	Nein
<b>Performance</b>	Schnell	Schnell
<b>Gespeicherte Prozeduren</b>	Ja	Ja

---

## Verbindung zur Datenbank mit PDO:

Der Verbindungsteil sieht unbeholfen aus, aber wir müssen damit umgehen. Der Rest des PDO ist einfach und nützlich und hilft auch dabei, das sichere Teil noch einfacher zu machen.

```
$connection = new PDO("mysql:host=localhost;dbname=myDatabase, username, password);
```

Die PDO-Verbindung erfolgt nach folgender Reihenfolge:

```
PDO ( database type : Host = host ; Datenbankname = database name , root , password );
```

---

## Sichere Auswahl aus der Datenbank mit PDO:

```
// We use a array to hold the data about whats the :var is in normal $var
$params = array(
    ':username' => '$username',
    ':email' => $mail,
);

// Prepare the SQL and using named secure parameters ":username"
$pdo->prepare('SELECT * FROM users WHERE username = :username AND email = :email');

// Execute the $params and send them to the $pdo->prepare
$pdo->execute($params);
```

Der Code, den Sie gerade gelesen haben, ist die geschützte Agenten-SQL-Injection

---

## Tutorials:

Installieren:

[Wie installiere ich PDO, wenn Sie es nicht haben?](#)

Führungen:

[W3Schools-Tutorial](#)

[Tuts + Tutorial \(Empfohlen\)](#)

## PDO-Einstellungsattribut-Fehlercode

PDO :: setAttribute legt ein Attribut für das Datenbank-Handle fest. Die Beschreibung von setAttribute lautet:

```
public bool PDO::setAttribute ( int $attribute , mixed $value )
```

**PDO :: ATTR\_ERRMODE:** Dieses Attribut wird für die Fehlerberichterstattung verwendet. Es kann einen der folgenden Werte haben.

- `PDO::ERRMODE_SILENT` : Wenn `ATTR_ERRMODE` nicht im Code festgelegt ist, ist `ERRMODE_SILENT` der Standardwert des Attributs `ATTR_ERRMODE`. Es setzt Fehlercodes. Im unbeaufsichtigten Modus gibt PDO keine Fehler aus, wenn in SQL ein Fehler auftritt. PDO gibt keine Warnungen aus. es wird einfach falsch zurückgegeben. Der Wert von `PDO :: ERRMODE_SILENT` ist 0. Das Skript wird ausgeführt, ohne dass ein Fehler oder eine Warnung generiert wird.
- `PDO::ERRMODE_WARNING` : Dieser Wert erhöht `E_WARNING`. Wenn im Warnmodus ein Fehler in SQL auftritt, gibt PDO Warnungen aus, das Skript wird jedoch weiterhin ausgeführt. Der Wert von `PDO :: ERRMODE_WARNING` ist 1. Das Skript wird mit der Warnung zum Fehler ausgeführt.
- `PDO::ERRMODE_EXCEPTION` : Dieser Wert `PDO::ERRMODE_EXCEPTION` Ausnahmen aus. Wenn im Ausnahmemodus ein Fehler in SQL auftritt, gibt PDO Ausnahmen aus, und das Skript wird nicht mehr ausgeführt. Der Wert von `PDO :: ERRMODE_EXCEPTION` ist 2. Das Skript beendet die Ausführung des Fehlers, der die Ausnahme auslöst.

Beispiel: Nun werden die verschiedenen Werte des Attributs `ATTR_ERRMODE` mit einigen Beispielen angezeigt. Erstellen Sie dazu eine Datenbank mit dem Namen `learn_project_db` und fügen Sie eine Tabelle mit dem Namen `user_table` . Das folgende SQL-Snippet kann dazu verwendet werden:

```
DROP DATABASE IF EXISTS `learn_project_db`;
CREATE DATABASE `learn_project_db`;
USE `learn_project_db`;
CREATE TABLE `user_table` (
  `user_email` varchar(50) PRIMARY KEY,
  `user_password` varchar(50) NOT NULL
);
INSERT INTO `user_table` (`user_email`, `user_password`) VALUES
('test1@example.com', '123'),
('test2@example.com', '1234'),
('test3@example.com', '12345');
```

Zuerst werden wir prüfen, was passiert, wenn wir `ATTR_ERRMODE` nicht setzen und in der SQL-Abfrage Fehler auftreten. Erstellen Sie eine PHP-Datei mit dem Namen `default.php` und versuchen Sie `default.php` :

```
<?php
  $server = "localhost";
  $db_username = "root";
  $db_password = "";
  $db_name = "learn_project_db";
  $conn = new PDO("mysql:host=$server;dbname=$db_name",$db_username,$db_password);
  $sql_query = "SELECT * FROM wrong_user_table";
  $stmt = $conn->prepare($sql_query);
  $stmt->execute();
  $result_set = $stmt->fetchAll();
  var_dump($result_set);
  /*Get the current error mode of PDO*/
  $current_error_mode = $conn->getAttribute(PDO::ATTR_ERRMODE);
```

```
echo "<br>";
echo "Value of PDO::ATTR_ERRMODE: ".$current_error_mode;
?>
```

Beachten Sie, dass der Tabellenname in der Abfrage `wrong_user_table` ist, die in der zuvor erstellten Datenbank nicht definiert ist. Da wir den `ATTR_ERRMODE` nicht festgelegt `ATTR_ERRMODE`, wird das Skript ausgeführt, ohne dass eine Ausnahme `ATTR_ERRMODE` oder eine Warnung `ATTR_ERRMODE` wird. Es wird ein leeres Array als Ergebnismenge ausgegeben. Wir sollten auch beachten, dass der Wert von `PDO::ATTR_ERRMODE` 0 ist.

Jetzt werden wir prüfen, was passiert, wenn wir `ATTR_ERRMODE` mit dem Wert `PDO::ERRMODE_WARNING` und ein Fehler in der SQL-Abfrage `PDO::ERRMODE_WARNING`. Erstellen Sie eine PHP-Datei mit dem Namen `warning.php` und versuchen Sie `warning.php`:

```
<?php
    $server = "localhost";
    $db_username = "root";
    $db_password = "";
    $db_name = "learn_project_db";
    $conn = new PDO("mysql:host=$server;dbname=$db_name",$db_username,$db_password);
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_WARNING);
    $sql_query = "SELECT * FROM wrong_user_table";
    $stmt = $conn->prepare($sql_query);
    $stmt->execute();
    $result_set = $stmt->fetchAll();
    var_dump($result_set);
    /*Get the current error mode of PDO*/
    $current_error_mode = $conn->getAttribute(PDO::ATTR_ERRMODE);
    echo "<br>";
    echo "Value of PDO::ATTR_ERRMODE: ".$current_error_mode;
?>
```

Die Ausgabe von `warning.php` lautet:

```
Warning: PDOStatement::execute(): SQLSTATE[42S02]: Base table or view not found: 1146 Table
'learn_project_db.wrong_user_table' doesn't exist in E:\xampp\htdocs\oop\db.php on line 10
array(0) { }
Value of PDO::ATTR_ERRMODE: 1
```

Dieses Mal, wenn wir den Wert `ATTR_ERRMODE` mit `PDO::ERRMODE_WARNING`, wird eine Warnmeldung `PDO::ERRMODE_WARNING`. Das Skript wird erfolgreich ausgeführt und zeigt als Ausgabe ein leeres Array mit dem Wert von `PDO::ATTR_ERRMODE` von 1.

Zum `ATTR_ERRMODE` prüfen wir, was passiert, wenn wir `ATTR_ERRMODE` mit dem Wert `PDO::ERRMODE_EXCEPTION` und in der SQL-Abfrage Fehler auftreten. Erstellen Sie eine PHP-Datei mit dem Namen `error.php` und versuchen Sie `error.php`:

```
<?php
    $server = "localhost";
    $db_username = "root";
    $db_password = "";
    $db_name = "learn_project_db";
    $conn = new PDO("mysql:host=$server;dbname=$db_name",$db_username,$db_password);
```

```
$conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
$sql_query = "SELECT * FROM wrong_user_table";
$stmt = $conn->prepare($sql_query);
$stmt->execute();
$result_set = $stmt->fetchAll();
var_dump($result_set);
/*Get the current error mode of PDO*/
$current_error_mode = $conn->getAttribute(PDO::ATTR_ERRMODE);
echo "<br>";
echo "Value of PDO::ATTR_ERRMODE: ".$current_error_mode;
?>
```

Die Ausgabe von `error.php` ist:

```
Fatal error: Uncaught exception 'PDOException' with message 'SQLSTATE[42S02]: Base table or
view not found: 1146 Table 'learn_project_db.wrong_user_table' doesn't exist' in
E:\xampp\htdocs\oop\db.php:10 Stack trace: #0 E:\xampp\htdocs\oop\db.php(10): PDOStatement-
>execute() #1 {main} thrown in E:\xampp\htdocs\oop\db.php on line 10
```

`ATTR_ERRMODE` **wir** `PDO::ERRMODE_EXCEPTION` **Wert** `ATTR_ERRMODE` **mit** `PDO::ERRMODE_EXCEPTION` , **wird** `PDOException` , **die** einen schwerwiegenden Fehler generiert. Das Skript wird nicht mehr ausgeführt, nachdem die Ausnahme ausgelöst wurde. Dies ist der gebräuchlichste Ansatz zum Behandeln von Fehlern in Bezug auf Datenbankabfragen. In den meisten `ATTR_ERRMODE` setzen wir das Attribut `ATTR_ERRMODE` mit diesem Wert, um alle Ausnahmen zu behandeln, die möglicherweise in der SQL-Abfrage vorhanden sind.

**Erste Schritte mit pdo online lesen:** <https://riptutorial.com/de/pdo/topic/4393/erste-schritte-mit-pdo>

---

# Kapitel 2: Bereiten Sie Ihre SQL-Anweisungen vor und führen Sie sie aus

## Bemerkungen

### Warnung

Die vorbereitete Anweisung kann sich nicht um einen Wildparameter für die Tabellennamen kümmern. Diese Aussage ist beispielsweise nicht korrekt:

```
$query = "SELECT name, city FROM ? WHERE id = ? AND country = ?";
```

Die korrekt vorbereitete Abfrage wäre:

```
$query = "SELECT name, city FROM users WHERE id = ? AND country = ?";
```

## Examples

### Verwendungszweck

```
// 1. Connect to the database (this example with MySQL)
$host = 'localhost';
$database = 'users';
$user = 'root';
$password = '';
$dsn = "mysql:host=$host;dbname=$database";
$pdo = new PDO($dsn, $user, $password);

// 2. Prepare your query

// 2.1 First way
$query_1 = "SELECT name, city FROM users WHERE id = ? AND country = ?";
$stmt_1 = $pdo->prepare($query_1);

// 2.2 Second way
$query_2 = "SELECT name, city FROM users WHERE id = :id AND country = :country";
$stmt_2 = $pdo->prepare($query_2);

// 3. Execute your query

// 3.1 With the first way
$stmt_1->execute([1, 'US']);

// 3.2 With the second way
$stmt_2->execute([
    ':id' => 1,
    ':country' => 'US'
]);

// 4. Fetch your data
```

```
$data_1 = $statement_1->fetchAll();  
$data_2 = $statement_2->fetchAll();
```

Bereiten Sie Ihre SQL-Anweisungen vor und führen Sie sie aus online lesen:

<https://riptutorial.com/de/pdo/topic/6511/bereiten-sie-ihre-sql-anweisungen-vor-und-fuehren-sie-sie-aus>

---

# Credits

S. No	Kapitel	Contributors
1	Erste Schritte mit pdo	<a href="#">arsho</a> , <a href="#">Community</a> , <a href="#">hjpotter92</a> , <a href="#">Ilyas Mimouni</a> , <a href="#">manian</a> , <a href="#">TheCrazyProfessor</a> , <a href="#">Your Common Sense</a>
2	Bereiten Sie Ihre SQL-Anweisungen vor und führen Sie sie aus	<a href="#">Anwar Nairi</a> , <a href="#">Your Common Sense</a>