

PDO-connect

MySQLi-connect is feitelijk alleen voor MySQL databases maar wat nu indien je het programma PHP wilt behouden en een andere database engine wilt benaderen?

PDO-connect is database onafhankelijk zodat je een minimale aanpassing hebt.



```
<?php
$dbh = new PDO('mysql:host=localhost;dbname=test', $user, $pass);
?>
```

`$dbh` = de verbinding naar de database

PDO-connect

```
<?php
// variabelen
$host = 'localhost';
$database='test';
$user = 'root';
$pass = '1234567';

$dbh = new PDO("mysql:host=$host;dbname=$database", $user, $pass);
?>
```

Om een verbinding te maken heb je nodig:

Host
Database
User
Password

\$dbh is een object ook wel DatabaseHandler genoemd

PDO-connect

```
<?php
// variabelen
$host = 'localhost';
$database='test';
$user = 'root';
$pass = '1234567';

$dbh = new PDO("mysql:host=$host;dbname=$database", $user, $pass);
?>
```

Met de verbinding ga je verbinden naar de tabel

```
$dbh->query('SELECT * from FOO')
```

```
$query = "SELECT * from FOO";
```

```
$dbh->query($query) // dit geeft een aantal rijen
```

PDO::query

PDO-connect

Try en Catch

Een database kan een connectie fout geven waardoor het programma 'hangt'.

```
<?php
try {
    $dbh = new PDO('mysql:host=localhost;dbname=test', $user, $pass);

    foreach($dbh->query('SELECT * from FOO') as $row) {
        print_r($row);
    }
    $dbh = null;
} catch (PDOException $e) {
    print "Error!: " . $e->getMessage() . "<br/>";
    die();
}
?>
```

PDO-connect

Foreach

Regel voor regel ophalen en verwerken

```
$dbh = new PDO('mysql:host=localhost;dbname=test', $user, $pass);  
  
foreach($dbh->query('SELECT * from FOO') as $row) {  
  
    print $row['voornaam'] . "\t";  
    print $row['achternaam'] . "\t";  
    print $row['telefoon'] . "\n";  
    print_r($row);  
}
```

PDO-connect

While

Is er wel iets gevonden, test op \$row blijf verwerken

```
$dbh = new PDO('mysql:host=localhost;dbname=test', $user, $pass);
```

```
$query="Select vn, an from users";
```

```
$qres = $dbh->prepare($query);
```

```
$qres->execute($query);
```

```
while ( $row = $qres->fetch() )    {  
    print_r($row);  
}
```

PDO-connect

For

In een lus afvragen van het resultaat

```
$dbh = new PDO('mysql:host=localhost;dbname=test', $user, $pass);

$qres = $dbh->prepare("select name FROM tbl_name");
$qres->execute();

For ( $i=0; $row = $qres->fetch(); $i++)      {

    echo $i." - ".$row['name']."<br/>";

}
```

PDO-connect

PDO variabelen doorgeven

Doorgeven van variabelen in een querystring

```
$dbh = new PDO('mysql:host=localhost;dbname=test', $user, $pass);
```

```
$t1 = 'voornaam';
```

```
$t2 = 'achternaam';
```

```
$t3 = 'email';
```

```
$qres = $dbh->query("SELECT * FROM table_users  
WHERE voornaam=? AND achternaam=? AND email=?",$t1, $t2, $t3);
```

```
while ($row = $qres->fetch())  
{  
    print_r($row);  
}
```


PDO-connect

PDO **setAttribute**

Opties die je kunt gebruiken

```
$dbh = new PDO('mysql:host=localhost;dbname=test', $user, $pass);
```

```
$dbh->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_SILENT);
```

PDO::ATTR_ERRMODE = foutmelding afhandeling

PDO::ERRMODE_SILENT = Geen foutmelding van de database

Dus geen melding op

```
print "Error!: " . $e->getMessage() . "<br/>";
```

PDO-connect

PDO **setAttribute**

Opties die je kunt gebruiken

```
$dbh = new PDO('mysql:host=localhost;dbname=test', $user, $pass);
```

```
$dbh->setAttribute(PDO::ATTR_TIMEOUT, 30);
```

Geef de database 30 seconden te tijd om met een resultaat te komen

```
$dbh->setAttribute(PDO::ATTR_PERSISTENT , TRUE);
```

Verbinding behouden en niet verbreken

PDO-connect

PDO attribute

Opties die je kunt gebruiken

```
PDO::ATTR_CASE  
PDO::ATTR_ERRMODE  
PDO::ATTR_ORACLE_NULLS  
PDO::ATTR_STRINGIFY_FETCHES  
PDO::ATTR_STATEMENT_CLASS  
PDO::ATTR_TIMEOUT  
PDO::ATTR_AUTOCOMMIT  
PDO::ATTR_EMULATE_PREPARES  
PDO::MYSQL_ATTR_USE_BUFFERED_QUERY  
PDO::ATTR_DEFAULT_FETCH_MODE
```

Bron:www.php.net