

Date  
19/02/2017

## PDO

Page No.	
Date:	

PHP Data Objects (PDO) is a lightweight system to access data from database with PHP. PDO has different drivers for different SQL database vendors. PDO provides a data-access abstraction layer, which means that, regardless of which database you are using, you use the same functions to issue queries and fetch data.

Thus PDO allows you to access a variety of different database and provides an easy way to query & retrieve results. PDO requires a minimum php version of 5.1.

```
$dbcon = new PDO('mysql:dbname=album;host=localhost; charset=utf8','root','');  
$dbcon->setAttribute(PDO::ATTR_EMULATE_PREPARE,  
S, false);  
$dbcon->setAttribute(PDO::ATTR_ERRMODE, PDO::ERR  
MODE_EXCEPTION);
```

```
{ $stmt = $dbcon->query('SELECT * FROM album');
```

(OR)

```
{ $stmt = $dbcon->prepare('SELECT * FROM album  
WHERE id = :id');
```

```
{ $stmt->execute(array('id' => 1));
```



in the form of associative array → FETCH\_ASSOC  
Associative array & numeric array → FETCH\_BOTH

Page No.	
Date:	

```
$rows = $stmt → fetchAll(PDO::FETCH_OBJ);  
// we can use fetch instead of fetchAll. fetch always  
// returns a single row of result if it more than 1 row.
```

Example: 'fetchAll' result:

Array

```
(  
    [0] => stdClass object  
    (  
        [id] => 1  
        [artist] => The Military Wives  
        [title] => In My Dreams  
    )  
)
```

'fetch' result:

stdClass object

```
(  
    [id] => 1  
    [artist] => The Military Wives  
    [title] => In My Dreams  
)
```

```
foreach ($rows as $row) {  
    echo '<br>'. 'Artist: ' . $row->artist;  
    echo '<br>'. 'Title: ' . $row->title;  
}
```



Date  
19/02/2017

## Error Handling

Page No.	
Date:	

We do not have to handle with try catch right away. We can catch it anytime that is appropriate. It may make more sense to catch it at a higher level like outside of the function that calls the PDO stuff.

```
function getData($db) {  
    $stmt = $db->query("SELECT * FROM album");  
    $stmt = $db->query("SELECT * FROM album");  
    return $stmt->fetchAll(PDO::FETCH_OBJ);  
}  
  
try {  
    getData($db);  
} catch (PDOException $ex) {  
    some_logging_function($ex->getMessage());  
}
```

We can hide the dangerous error messages in production by turning display\_errors off and just reading your error log.

Reference: [wiki.hashphp.org/PDO-Tutorial-for-Mysql-Developers](http://wiki.hashphp.org/PDO-Tutorial-for-Mysql-Developers)  
[php.net/manual/en/intro.pdo.php](http://php.net/manual/en/intro.pdo.php)

Date  
19/02/2017

## Transactions

Page No.	
Date	

Here is an example of using transactions in PDO.  
calling `beginTransaction()` turns off auto commit automatically.

```
try {  
    $db->beginTransaction();  
    $db->exec("SOME QUERY");  
  
    $stmt = $db->prepare("SOME OTHER QUERY ?");  
    $stmt->execute(array($value));  
  
    $stmt = $db->prepare("YET ANOTHER QUERY ??");  
    $stmt->execute(array($value2, $value3));  
  
    $db->commit();  
} catch (PDOException $ex) {  
    // something went wrong rollback!  
    $db->rollBack();  
    echo $ex->getMessage();  
}
```



Date  
21/02/2017

## utf8mb4 instead of utf8 only

Specify the utf8mb4 character set on all tables and text columns in your database. This makes MySQL physically store & retrieve values encoded natively in UTF-8. Note that MySQL will implicitly use utf8mb4 encoding if a utf8mb4\_\* collation is specified (without any explicit character set).

In older versions of MySQL (< 5.5.3), we ~~could~~ <sup>all</sup> unfortunately be forced to use simply utf8, which only supports a subset of Unicode characters. I wish I were kidding.



Date  
27/02/2017

## Dump database with Unix shell

Page No.	
Date:	

Dump database db1:

```
mysqldump -h localhost -u root -p db1 > db1bkp.sql
```

Dump database db1's only 3 tables tb1, tb2 & tb3:

```
mysqldump -h localhost -u root -p db1 tb1 tb2 tb3 >  
dbtablebkp.sql
```

Dump only 3 databases db1, db2 & db3:

```
mysqldump -h localhost -u root -p --databases db1  
db2 db3 > somebackup.sql
```

Dump all databases:

```
mysqldump -h localhost -u root -p --all-databases >  
dump.sql
```

Note: The `--databases` option causes all names on the command line to be treated as database names. Without this option, `mysqldump` treats the first name as a database name & those following as table names.

Note: For the most 1st command on this page, if we not use `--databases` then the created backup sql file i.e. `db1bkp.sql` will not contain 'create database' & 'use database' commands.

Remember, all these commands will when you are not logged to MySQL server.



Date  
27/02/2019

Restore database (or database table)  
from backup


If bkp.sql contains create table & use database commands already inside it:

```
mysql -u root -p < bkp.sql
```

Otherwise:

```
mysql -u root -p databasename < bkp.sql
```



first we will need to create a db with name 'database name'

Some MySQL commands with unix shell

Login in MySQL:

```
mysql -h localhost -u root -p
```

Create a database:

```
create database databasename;
```

View all databases:

```
show databases;
```

Select or use a database:

```
use databasename;
```

View all tables from a database of mysql server:

```
show tables;
```

Description of a table:

```
describe tablename; or desc tablename;
```

Delete a database from mysql server:

```
drop database databasename;
```

Delete a table:

```
drop table tablename;
```



Date  
29/04/2017

## MySQL Part - 1

Page No.	
Date:	

Create Table command:

```
create table <table name> (<column name 1> <data type> (<size>), <column name 2> <data type> (<size>));
```

Example:

```
create table 'user'(  
  'user_id' int(10) unsigned not null auto_increment  
  comment 'User Id',  
  'name' varchar(32) default null comment 'Name',  
  'created' timestamp not null default current-time  
  stamp comment 'User Created Time',  
  'modified' timestamp not null default current-tim  
  estamp on update current_timestamp comment 'Mod Time',  
  'is_active' smallint(6) not null default '1' comment  
  'User is Active',  
  primary key ('user_id'),  
  unique key 'user_name' ('name')  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COMMENT='Admin  
User Table';
```



Date  
12/04/2019

prestashop, opencart, Joomla, wordpress

Page No.	
Date	12/04/2019

## Naming database, table, column

As a standard naming convention (as from Magento), always use small letters & underscore to name database, table & columns.

Always name primary key column name - table name - id. For example - For table 'user', primary key column name will be 'user-id' (as from Magento). Benefit of this type of naming, when we use this as foreign key in another table, it would be easier to understand.

## Download & Install "SQLyog community"

Just type "SQLyog community" & search on google. Click on first link:

<https://github.com/webbyog/sqlযোগ-community/wiki/Downloads>

and now download and install the same.