Dept of Computer Science
Ahmadu Bello University

WEB APPLICATION ENGINEERING II

Lecture #10

Umar Ibrahim Enesi

Dept of Computer Science
Ahmadu Bello University

Objectives

- Gain understanding on using PDO

Dept of Computer Science
Ahmadu Bello University

What is PDO

- PDO- **P**HP **D**ata **O**bjects
- PDO is a PHP extension (not PHP's core)
- Formalizes access to databases
- Not a database abstraction layer – provides a data-access abstraction layer
- Ships with PHP
- Supports various databases (MySQL, Oracle database, SQLite…)

Dept of Computer Science
Ahmadu Bello University

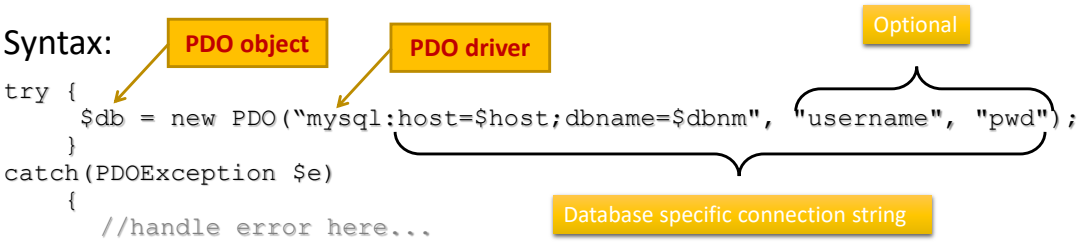
Checking for & Enabling PDO

- There is a database-specific PDO driver for every database that supports PDO
- Use `phpinfo()` to check the status of PDO and availability of database-specific PDO driver.
- Or programmatically, `PDO::getAvailableDrivers()`
- To enable PDO simply configure the appropriate extension in `php.ini`
- On windows just uncomment

Dept of Computer Science
Ahmadu Bello University

Connecting to Database (MySQL)

Syntax:



- Instead of using `try{}catch(){}`, you might want to handle the exception in an application global exception handler

Dept of Computer Science
Ahmadu Bello University

Connecting to Database (Persistence)

Syntax:

```
try {
      $db = new PDO(
                  "mysql:host=$host;dbname=$dbnm",
                  "username",
                  "pwd",
                  array( PDO::ATTR_PERSISTENT => true )
                  );
    }
catch(PDOException $e)
    {
      //handle error here...
    }
```

***Note**: Using `PDO::setAttribute()` instead of array format after connection has been made will not have any effect*

Dept of Computer Science
Ahmadu Bello University

Closing Connection

- By default, non-persistent connections are implicitly closed after a script finish executing.
- Explicitly,

```
$db = null;
```

- Avoid this,

```
$db_copy = $db;
$db = null;
```

Dept of Computer Science
Ahmadu Bello University

Querying Databases: SELECT

- Simplistically:

```
$result = $db->query($select_sql);
```

- On success, returns a `PDOStatement` object that can be iterated using foreach...loop.
- On failure, false is returned
- `$select_sql` should be properly escaped
- Call `$result->closeCursor()` before issuing another SELECT query.
- Use `fetch()` or `fetchAll()` to get result of the query

Dept of Computer Science
Ahmadu Bello University

Querying Databases: FETCH Modes

- `query()` returns `PDOStatement` object on success.
- By default `PDOStatement` object uses the `PDO::FETCH_BOTH` mode
- Use `setFetchMode()` to change the default fetch mode
- Most common modes:
  - `PDO::FETCH_ASSOC`: Table column name as keys
  - `PDO::FETCH_NUM`: Integer index as keys in table column order
  - `PDO::FETCH_BOTH`: A combination of column names and integer index (always double the size of `FETCH_ASSOC` or `FETCH_NUM`)

Dept of Computer Science
Ahmadu Bello University

Querying Databases: INSERT, UPDATE & DELETE

Syntax:

```
$count = $db->exec($sql);
```

- On success returns the number of rows affected (integer)
- On failure returns false
- Use (===) to make sure 0 is converted to false
- `$sql` should be properly escaped

Dept of Computer Science
Ahmadu Bello University

Querying Databases: Prepared Statements

- Prepared statement is like an SQL template
- Can be used with different variable parameters.
- It improves performance
- Optimize resources
- Prevent SQL injection (automatic escaping)
- It is more convenient when issuing the same query multiple times
- Can be used for SELECT, INSERT, UPDATE and DELETE

Dept of Computer Science
Ahmadu Bello University

Querying Databases: Prepared Statements

Syntax 1 (binding with named parameter):

```
$stmt = $dbh->prepare("INSERT INTO tbluser(username, pwd)
      VALUES (:usn, :pwd)");
$stmt->bindParam(':usn', $usn, PDO::PARAM_STR);
$stmt->bindParam(':pwd', $pwd, PDO::PARAM_STR, 10);
$usn= 'umar';
$pwd= 'enesi_pass';
$stmt->execute();
```

- Recall that `PDOStatement` has methods for fetching result set and to enquire how many rows are affect by its last execution.

Dept of Computer Science
Ahmadu Bello University

Querying Databases: Prepared Statements

Syntax 2 (binding with positional parameter):

```
$stmt = $dbh->prepare("INSERT INTO tbluser(username, pwd)
      VALUES (?, ?)");
$stmt->bindParam(1, $usn, PDO::PARAM_STR);
$stmt->bindParam(2, $pwd, PDO::PARAM_STR, 10);
$usn= 'umar';
$pwd= 'enesi_pass';
$stmt->execute();
```

Dept of Computer Science
Ahmadu Bello University

Querying Databases: Prepared Statements

Syntax 3 (using array of parameter values):

```
$stmt = $dbh->prepare("INSERT INTO tbluser(username, pwd)
      VALUES (:usn, :pwd)");
$usn= 'umar';
$pwd= 'enesi_pass';
$vals = array(":usn"=>$usn, ":pwd"=>$pwd);
$stmt->execute($vals);
```

- All array values will be mapped as `PDO::PARAM_STR`

Dept of Computer Science
Ahmadu Bello University

Querying Databases: Prepared Statements

Syntax 4 (using array of positional values):

```
$stmt = $dbh->prepare("INSERT INTO tbluser(username, pwd)
      VALUES (?, ?)");
$usn= 'umar';
$pwd= 'enesi_pass';
$vals = array($usn, $pwd);
$stmt->execute($vals);
```

Dept of Computer Science
Ahmadu Bello University

Querying Databases: Prepared Statements

Syntax 5 (Special case with IN clause):

```
$params= array('umar', 'ibrahim', 'enesi');
$place_holders = implode(',', array_fill(0, count($params), '?'));
$stmt = $dbh->prepare("SELECT * FROM tbluser WHERE username
      IN ($place_holders)");
$stmt->execute($params);
```

Dept of Computer Science
Ahmadu Bello University

Transactions

- By default auto-commit mode is on
- PDO transactions turns auto-commit mode off
- Thus one has to explicitly commit queries
- This allows for one to query the database multiple times before committing

Dept of Computer Science
Ahmadu Bello University

Transactions

General Syntax:

```
$db -> beginTransaction();
//perform multiple queries…
$db->commit();
```

To cancel a transaction call:

```
$db->rollback();
```

During a transaction, the id of the last INSERT can be retrieved:

```
$lasted = $db -> lastInsertId();
```

Dept of Computer Science
Ahmadu Bello University

Error Handling

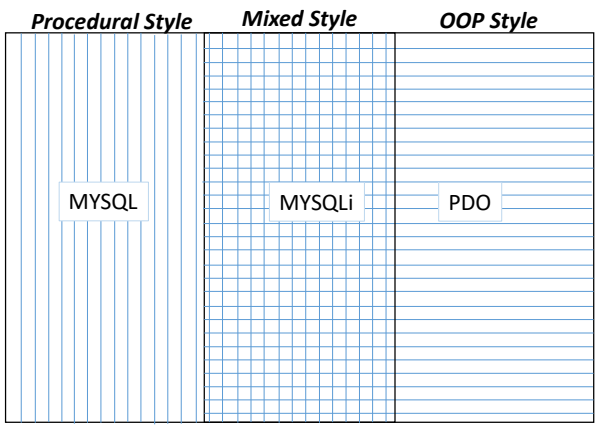
- Exceptions may occur during database connection
- But errors can occur during query execution
  - Invalid SQL statement
  - Insufficient privileges
  - Error in PHP logic
- To catch and handle all errors from the point of connection to the point of disconnecting from the database;

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

- This will cause Exception to be through in case of any error.

Dept of Computer Science
Ahmadu Bello University

Summary on Database Access Libraries



Dept of Computer Science
Ahmadu Bello University

Reference

- http://www.phpro.org/tutorials/Introduction-to-PHP-PDO.html#4
- PHP Manual
- Wrox Professional PHP6