PHP Connection with MySQL Database

Introduction

Higher version of PHP may work with MySQL database in two ways.

(1) By using MySQLi extension

(2) By using PDO (PHP Data Object)

Here following examples are given for PDO as it may work with 12 different databases.

Establish a Connection

```
<?php
   $servername = "localhost";
   $username = "username";  // it can be “root”
   $password = "password";   // it can be “root” or “”

   try {
      $conn = new PDO("mysql:host=$servername;dbname=myDB", $username,
   $password);  // my DB is database name
      // set the PDO error mode to exception
      $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
      echo "Connected successfully";
      }
   catch(PDOException $e)
      {
      echo "Connection failed: " . $e->getMessage();
      }
   ?>
```

To Close the Connection

```
$conn = null;
```

To Create a Database

```
<?php
 $servername = "localhost";
 $username = "username";
 $password = "password";

 try {
    $conn = new PDO("mysql:host=$servername", $username, $password);
    // set the PDO error mode to exception
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    $sql = "CREATE DATABASE myDBPDO";
    // use exec() because no results are returned
    $conn->exec($sql);
    echo "Database created successfully<br>";
    }
 catch(PDOException $e)
    {
    echo $sql . "<br>" . $e->getMessage();
    }

 $conn = null;
 ?>
```

To Create a Table

```
<?php
    $servername = "localhost";
    $username = "username";
    $password = "password";
    $dbname = "myDBPDO";

    try {
        $conn = new
    PDO("mysql:host=$servername;dbname=$dbname",
    $username, $password);
        // set the PDO error mode to exception
        $conn->setAttribute(PDO::ATTR_ERRMODE,
    PDO::ERRMODE_EXCEPTION);
```

```
// sql to create table
        $sql = "CREATE TABLE MyGuests (
        id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
        firstname VARCHAR(30) NOT NULL,
        lastname VARCHAR(30) NOT NULL,
        email VARCHAR(50),
        reg_date TIMESTAMP
        )";

        // use exec() because no results are returned
        $conn->exec($sql);
        echo "Table MyGuests created successfully";
        }
    catch(PDOException $e)
        {
        echo $sql . "<br>" . $e->getMessage();
        }

    $conn = null;
    ?>
```

To Insert a Data

```
<?php
 $servername = "localhost";
 $username = "username";
 $password = "password";
 $dbname = "myDBPDO";

 try {
    $conn = new PDO("mysql:host=$servername;dbname=$dbname", $username, $password);
    // set the PDO error mode to exception
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    $sql = "INSERT INTO MyGuests (firstname, lastname, email)
    VALUES ('John', 'Doe', 'john@example.com')";
    // use exec() because no results are returned
    $conn->exec($sql);
    echo "New record created successfully";
    }
 catch(PDOException $e)
    {
    echo $sql . "<br>" . $e->getMessage();
    }

 $conn = null;
 ?>
```

To Insert Multiple Data

```
<?php
   $servername = "localhost";
   $username = "username";
   $password = "password";
   $dbname = "myDBPDO";

   try {
      $conn = new PDO("mysql:host=$servername;dbname=$dbname", $username, $password);
      // set the PDO error mode to exception
      $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

      // begin the transaction
      $conn->beginTransaction();
      // our SQL statements
      $conn->exec("INSERT INTO MyGuests (firstname, lastname, email)
      VALUES ('John', 'Doe', 'john@example.com')");
```

```
$conn->exec("INSERT INTO MyGuests (firstname, lastname, email)
    VALUES ('Mary', 'Moe', 'mary@example.com')");
    $conn->exec("INSERT INTO MyGuests (firstname, lastname, email)
    VALUES ('Julie', 'Dooley', 'julie@example.com')");

    // commit the transaction
    $conn->commit();
    echo "New records created successfully";
    }
  catch(PDOException $e)
    {
    // roll back the transaction if something failed
    $conn->rollback();
    echo "Error: " . $e->getMessage();
    }

  $conn = null;
  ?>
```

To Insert Data with Prepared Statement

```
<?php
   $servername = "localhost";
   $username = "username";
   $password = "password";
   $dbname = "myDBPDO";

   try {
      $conn = new PDO("mysql:host=$servername;dbname=$dbname", $username, $password);
      // set the PDO error mode to exception
      $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

      // prepare sql and bind parameters
      $stmt = $conn->prepare("INSERT INTO MyGuests (firstname, lastname, email)
      VALUES (:firstname, :lastname, :email)");
      $stmt->bindParam(':firstname', $firstname);
      $stmt->bindParam(':lastname', $lastname);
      $stmt->bindParam(':email', $email);
```

```
// insert a row
    $firstname = "John";
    $lastname = "Doe";
    $email = "john@example.com";
    $stmt->execute();

    // insert another row
    $firstname = "Mary";
    $lastname = "Moe";
    $email = "mary@example.com";
    $stmt->execute();

    // insert another row
    $firstname = "Julie";
    $lastname = "Dooley";
    $email = "julie@example.com";
    $stmt->execute();

    echo "New records created successfully";
    }
 catch(PDOException $e)
    {
    echo "Error: " . $e->getMessage();
    }
 $conn = null;
 ?>
```

To Select Data

```
<?php
 echo "<table style='border: solid 1px black;'>";
 echo "<tr><th>Id</th><th>Firstname</th><th>Lastname</th></tr>";

 class TableRows extends RecursiveIteratorIterator {
    function __construct($it) {
       parent::__construct($it, self::LEAVES_ONLY);
    }

    function current() {
       return "<td style='width:150px;border:1px solid black;'>" . parent::current(). "</td>";
    }

    function beginChildren() {
       echo "<tr>";
    }

    function endChildren() {
       echo "</tr>" . "\n";
    }
 }
```

```
$servername = "localhost";
    $username = "username";
    $password = "password";
    $dbname = "myDBPDO";

    try {
        $conn = new PDO("mysql:host=$servername;dbname=$dbname", $username, $password);
        $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
        $stmt = $conn->prepare("SELECT id, firstname, lastname FROM MyGuests");
        $stmt->execute();

        // set the resulting array to associative
        $result = $stmt->setFetchMode(PDO::FETCH_ASSOC);
        foreach(new TableRows(new RecursiveArrayIterator($stmt->fetchAll())) as $k=>$v) {
            echo $v;
        }
    }
    catch(PDOException $e) {
        echo "Error: " . $e->getMessage();
    }
    $conn = null;
    echo "</table>";
    ?>
```

To Delete a Data

```
<?php
  $servername = "localhost";
  $username = "username";
  $password = "password";
  $dbname = "myDBPDO";

  try {
    $conn = new PDO("mysql:host=$servername;dbname=$dbname", $username, $password);
    // set the PDO error mode to exception
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

    // sql to delete a record
    $sql = "DELETE FROM MyGuests WHERE id=3";

    // use exec() because no results are returned
    $conn->exec($sql);
    echo "Record deleted successfully";
    }
  catch(PDOException $e)
    {
    echo $sql . "<br>" . $e->getMessage();
    }

  $conn = null;
  ?>
```

To Update a Data

```
<?php
   $servername = "localhost";
   $username = "username";
   $password = "password";
   $dbname = "myDBPDO";

   try {
      $conn = new PDO("mysql:host=$servername;dbname=$dbname", $username, $password);
      // set the PDO error mode to exception
      $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

      $sql = "UPDATE MyGuests SET lastname='Doe' WHERE id=2";

      // Prepare statement
      $stmt = $conn->prepare($sql);

      // execute the query
      $stmt->execute();

      // echo a message to say the UPDATE succeeded
      echo $stmt->rowCount() . " records UPDATED successfully";
      }
   catch(PDOException $e)
      {
      echo $sql . "<br>" . $e->getMessage();
      }

   $conn = null;
   ?>
```

Reference

- https://www.w3schools.com/php/