Integrating PHP with MySQL

Chencha Jacob

July 2, 2015

Available Options

1. mysql: The original API
2. mysqli: A better version of the original. The *i* stands for *improved*‘
3. PDO: PHP Data Objects

Comparison chart

mysqli OOP

Procedural	Object-oriented
`mysqli_connect`	`$mysqli = new mysqli`
`mysqli_connect_errno`	`$mysqli->connect_errno`
`mysqli_connect_error`	`$mysqli->connect_error`
`mysqli_real_escape_string`	`$mysqli->real_escape_string`
`mysqli_query`	`$mysqli->query`
`mysqli_fetch_assoc`	`$mysqli->fetch_assoc`
`mysqli_close`	`$mysqli->close`

Note they are almost similar. We will be switching the methods to OOP soon enough.

Steps in connecting to DB

1. Create a database connection: This is where we authenticate ourselves to the *MySQL* server
2. Perform a database query: Here we pass data using the usual *SQL* syntax
3. Act on output of the query: For example show the shopper items on their cart
4. Release returned data: This is a general good practise for large datasets in memory
5. Close the connection: MySQL connection is an expensive resource, close it when done.

Connecting to the database

```php
<?php
// 1. Create a database connection
$dbhost = "localhost";
$dbuser = "test_app_user";
$dbpass = "test_app_secret";
$dbname = "test_app";
$connection = mysqli_connect($dbhost, $dbuser, $dbpass, $dbname);
// Test if connection occurred.
if(mysqli_connect_errno()) {
        die("Database connection failed: " .
                        mysqli_connect_error() .
                        " (" . mysqli_connect_errno() . ")"
           );
}

// 5. Close database connection
mysqli_close($connection);
?>
```

- The connection variable is the handle that will from now on use to interact with the database
- MySQL connection is not guaranteed to succeed so its always good to confirm before.
- Some reasons why MySQL connection may fail include:
- MySQL service is not running
- Unix socket file maybe missing
- Server maybe overloaded
- MySQL not connected via normal port
- etc

Querying the database

```php
<?php
// 1. Create a database connection
$dbhost = "localhost";
$dbuser = "test_app_user";
$dbpass = "test_app_secret";
$dbname = "test_app";
$connection = mysqli_connect($dbhost, $dbuser, $dbpass, $dbname);
// Test if connection succeeded
if(mysqli_connect_errno()) {
        die("Database connection failed: " .
                        mysqli_connect_error() .
                        " (" . mysqli_connect_errno() . ")"
           );
}
// 2. Perform database query
$query  = "SELECT * FROM subjects ";
$result = mysqli_query($connection, $query);
// Test if there was a query error
if (!$result) {
        die("Database query failed.");
}

//var_dump($result);die();

// 3. Use returned data (if any)
while($row = mysqli_fetch_row($result)) {
        // output data from each row
        var_dump($row);
}
// 4. Release returned data
mysqli_free_result($result);
// 5. Close database connection
mysqli_close($connection);
?>
```

- Try modifying the SQL query, what do you get?
- We use while since we don't have the ability to manually change the results pointer. See the dumped results

More complex query

```php
<?php
// 1. Create a database connection
$dbhost = "localhost";
$dbuser = "test_app_user";
$dbpass = "test_app_secret";
$dbname = "test_app";
$connection = mysqli_connect($dbhost, $dbuser, $dbpass, $dbname);
// Test if connection succeeded
if(mysqli_connect_errno()) {
        die("Database connection failed: " .
                        mysqli_connect_error() .
                        " (" . mysqli_connect_errno() . ")"
           );
}
// 2. Perform database query
$query  = "SELECT * ";
$query .= "FROM subjects ";
$query .= "WHERE visible = 1 ";
$query .= "ORDER BY position ASC";
$result = mysqli_query($connection, $query);
// Test if there was a query error
if (!$result) {
        die("Database query failed.");
}

//var_dump($result);die();
// 3. Use returned data (if any)
while($row = mysqli_fetch_row($result)) {
        // output data from each row
        var_dump($row);
}
// 4. Release returned data
mysqli_free_result($result);
// 5. Close database connection
mysqli_close($connection);
?>
```

- We can split up the query into multiple fields. This moves helps readability.

Options when fetching data

mysqli_fetch_row

This provides the results in a standard array that is key indexed

```
array(4) {
         [0] =>
                  string(1) "1"
                  [1] =>
                  string(17) "About Widget Corp"
                  [2] =>
                  string(1) "1"
                  [3] =>
                  string(1) "1"
```

mysqli_fetch_assoc

This provides results as an associative array which is quite useful for key value pairs.

```
array(4) {
        'id' =>
                string(1) "1"
                'menu_name' =>
                string(17) "About Widget Corp"
                'position' =>
                string(1) "1"
                'visible' =>
                string(1) "1"
}
```

mysqli_fetch_array

Provides results in either or both

```
array(8) {
        [0] =>
                string(1) "1"
                'id' =>
                string(1) "1"
                [1] =>
                string(17) "About Widget Corp"
                'menu_name' =>
                string(17) "About Widget Corp"
                [2] =>
                string(1) "1"
                'position' =>
                string(1) "1"
                [3] =>
                string(1) "1"
                'visible' =>
                string(1) "1"
}
```

Usage results

```
<?php
        // 1. Create a database connection
        $dbhost = "localhost";
$dbuser = "test_app_user";
$dbpass = "test_app_secret";
$dbname = "test_app";
$connection = mysqli_connect($dbhost, $dbuser, $dbpass, $dbname);
// Test if connection succeeded
if(mysqli_connect_errno()) {
        die("Database connection failed: " .
                        mysqli_connect_error() .
                        " (" . mysqli_connect_errno() . ")"
           );
}
        // 2. Perform database query
$query  = "SELECT * ";
$query .= "FROM subjects ";
$query .= "WHERE visible = 1 ";
$query .= "ORDER BY position ASC";
$result = mysqli_query($connection, $query);
// Test if there was a query error
if (!$result) {
        die("Database query failed.");
}
?>


        <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
        "http://www.w3.org/TR/html4/loose.dtd">

        <html lang="en">
        <head>
        <title>Databases</title>
        </head>
        <body>

        <ul>
        <?php
        // 3. Use returned data (if any)
        while($subject = mysqli_fetch_assoc($result)) {
                // output data from each row
                ?>
                        <li><?php echo $subject["menu_name"] . " (" . $subject["id"]
                        <?php
```

```
        }
?>
        </ul>

        <?php
        // 4. Release returned data
        mysqli_free_result($result);
?>
        </body>
        </html>

        <?php
        // 5. Close database connection
        mysqli_close($connection);
?>
```

Other results from query

	success	failure
SELECT	resource	false
INSERT	true	false
UPDATE	true	false
DELETE	true	false

Sample insertion

```
        <?php
        // 1. Create a database connection
        $dbhost = "localhost";
$dbuser = "test_app_user";
$dbpass = "test_app_secret";
$dbname = "test_app";
$connection = mysqli_connect($dbhost, $dbuser, $dbpass, $dbname);
// Test if connection succeeded
if(mysqli_connect_errno()) {
        die("Database connection failed: " .
                        mysqli_connect_error() .
                        " (" . mysqli_connect_errno() . ")"
           );
}
// Often these are form values in $_POST
$menu_name = "Edit me";
$position = 4;
$visible = 1;

// 2. Perform database query
$query  = "INSERT INTO subjects (";
$query .= "  menu_name, position, visible";
$query .= ") VALUES (";
$query .= "  '{$menu_name}', {$position}, {$visible}";
$query .= ")";

$result = mysqli_query($connection, $query);

if ($result) {
        // Success
        // Do post sucess processing
        echo "Success!";
} else {
        // Failure
        // Do post failure processing, maybe notify the admin?
        die("Database query failed. " . mysqli_error($connection));
}
// 5. Close database connection
mysqli_close($connection);
?>
```

- Note positioning of the various columns

- Post insertion we usually carry out another activity such as:
- Redirect user to success page
- Download digital asset
- To find out the *id* of the record we just inserted, we can use *mysqli_insert_id()*

```
Usage:

    $id=mysqli_insert_id($connection);

- This *id* would be useful for post success/failure processing
```

Sample update

```
        <?php
        // 1. Create a database connection

        $dbhost = "localhost";
$dbuser = "test_app_user";
$dbpass = "test_app_secret";
$dbname = "test_app";
$connection = mysqli_connect($dbhost, $dbuser, $dbpass, $dbname);
// Test if connection succeeded
if(mysqli_connect_errno()) {
        die("Database connection failed: " .
                        mysqli_connect_error() .
                        " (" . mysqli_connect_errno() . ")"
           );
}
// Often these are form values in $_POST
$id = 5;
$menu_name = "Delete me";
$position = 4;
$visible = 1;

// 2. Perform database query
$query  = "UPDATE subjects SET ";
$query .= "menu_name = '{$menu_name}', ";
$query .= "position = {$position}, ";
$query .= "visible = {$visible} ";
$query .= "WHERE id = {$id}";

$result = mysqli_query($connection, $query);


if ($result) {
        assert(mysqli_affected_rows($connection) == 1);
        // Success
        // redirect_to("somepage.php");
        echo "Success!";
} else {
        // Failure
        // $message = "Subject update failed";
        die("Database query failed. " . mysqli_error($connection));
}
?>
```

- We must provide an *id* otherwise *MySQL* will overwrite ALL columns

- Where you put your *id* will be dictated by the architecture of your application but it is usually encoded in either:
- The url eg *myapp.com/users/1234/profile* The *id* here would be 1234
- In cookie as such accessed `$_COOKIE['user_id']`
- In session as such accessed `$_SESSION['user_id']`
- We note that the *UPDATE* action will still return *true* even for a null set. In this cases we need to run `mysqli_affected_rows($connection)` to check how many rows we're affected and assert that it is the number we expected.
- If we pass the exact same data then MySQL doesn't do anything and affected rows count is 0
- Try updating other columns
- Try seeing the results from the MySQL shell

Sample delete

```php
<?php
$dbhost = "localhost";
$dbuser = "test_app_user";
$dbpass = "test_app_secret";
$dbname = "test_app";
  // 1. Create a database connection
  $connection = mysqli_connect($dbhost, $dbuser, $dbpass, $dbname);
  // Test if connection succeeded
  if(mysqli_connect_errno()) {
    die("Database connection failed: " .
         mysqli_connect_error() .
         " (" . mysqli_connect_errno() . ")"
    );
  }
    // Often these are form values in $_POST
    $id = 5;

    // 2. Perform database query
    $query  = "DELETE FROM subjects ";
    $query .= "WHERE id = {$id} ";
    $query .= "LIMIT 1";

    $result = mysqli_query($connection, $query);

    if ($result && mysqli_affected_rows($connection) == 1) {
        // Success
        // redirect_to("somepage.php");
        echo "Success!";
    } else {
        // Failure
        // $message = "Subject delete failed";
        die("Database query failed. " . mysqli_error($connection));
    }
  // 5. Close database connection
  mysqli_close($connection);
?>
```

- We can limit the number of deletions just to be safe.
- To be safer you can carry out a *SELECT* query first then do the delete

Assignment

Aggregrate the various scripts to come up with a reusable function for all *CRUD* methods ie impliment the methods below

- function read(*connection*,query)
- function create(*connection*,columns,$data)
- function update(*connection*,id,*columns*,data)
- function delete(*connection*,id)