PHP Access File, Database and Sqlite

Fajiang Yu, fjyu@whu.edu.cn

School of Computer, Wuhan University

2017.4

Agenda

1. Access File
2. Access MySql
3. Access Sqlite

Agenda

1. Access File

2. Access MySql

3. Access Sqlite

handle with care

There are three distinct steps to be followed

- Open the file and assign it a file handle
- Interact with the file, via its handle, and extract its contents into a PHP variable
- Close the file

access file

```php
<?php
   // set file to read
   $file = '/usr/local/omelette.txt'
            or die('Could not open file!');
   // open file
   $fh = fopen($file, 'r') or die('Could not open file!');
   // read file contents
   $data = fread($fh, filesize($file))
           or die('Could not read file!');
   // close file
   fclose($fh);
   // print file contents
   echo $data;
 ?>
```

open mode

There are three distinct steps to be followed

- 'r' – opens a file in read mode
- 'w' – opens a file in write mode, destroying existing file contents
- 'a' – opens a file in append mode, preserving existing file contents

read file into array

```
<?php
   // set file to read
   $file = '/usr/local/omelette.txt'
            or die('Could not read file!');
   // read file into array
   $data = file($file) or die('Could not read file!');
   // loop through array and print each line
   foreach ($data as $line) {
      echo $line;
   }
 ?>
```

read file into string

```
<?php
   // set file to read
   $file = '/usr/local/omelette.txt' ;
   // read file into string
   $data = file_get_contents($file)
            or die('Could not read file!');
   // print contents
   echo $data;
 ?>
```

include file

header.php

```
<html>
<head>
<title><?php echo $page['title'];?></title>
</head>
<body>
<!-- top menu bar -->
<table width="90%" border="0" cellspacing="5" cellpadding="5">
<tr>
<td><a href="#">Home</a></td>
<td><a href="#">Site Map</a></td>
<td><a href="#">Search</a></td>
<td><a href="#">Help</a></td>
</tr>
</table>
<!-- header ends -->
```

include file

footer.php

```
<!-- footer begins -->
<br />
<center>
Your usage of this site is subject to its published
<a href="tac.html">terms and conditions</a>.
Data is copyright Big Company Inc, 1995-
<?php echo date("Y", mktime()); ?>
</center>
</body>
</html>
```

include file

```
<?php
   // create an array to set page-level variables
   $page = array();
   $page['title'] = 'Product Catalog';
   /* once the file is imported, the variables
      set above will become available to it */
   // include the page header
   include('header.php');
 ?>
   <!-- HTML content here -->
<?php
   // include the page footer
   include('footer.php');
 ?>
```

write file

```
<?php
   // set file to write
   $file = '/tmp/dump.txt';
   // open file
   $fh = fopen($file, 'w') or die('Could not open file!');
   // write (binary) to file
   fwrite($fh, "Look, Ma, I wrote a file!")
         or die('Could not write to file');
   // close file
   fclose($fh);
 ?>
```

write file

```
<?php
   // set file to write
   $filename = '/tmp/dump.txt';

   // write to file
   file_put_contents($filename,
                    "Look, Ma, I wrote a file!")
      or die('Could not write to file');
 ?>
```

file information

```
<html>
<head></head>

<body>
<?php
   /* if form has not yet been submitted, display input box */
   if (!isset($_POST['file'])) {
 ?>
      <form action=
            "<?php echo $_SERVER['PHP_SELF']; ?>" method="post">
         Enter file path <input type="text" name="file">
      </form>
<?php
   }
   // else process form input
   else {
      echo 'File name: <b>'.$_POST['file']  .'</b><br />';
```

file information

```
/* check if file exists and display appropriate message */
if (file_exists($_POST['file'])) {
   // print file size
   echo 'File size: '.filesize($_POST['file'])
        .' bytes<br />';
   // print file owner
   echo 'File owner: '.fileowner($_POST['file']).'<br />';
   // print file group
   echo 'File group: '.filegroup($_POST['file']).'<br />';
   // print file permissions
   echo 'File permissions: '.fileperms($_POST['file'])
        .'<br />';
   // print file type
   echo 'File type: '.filetype($_POST['file']).'<br />';
   // print file last access time
   echo 'File last accessed on: '
        .date('Y-m-d', fileatime($_POST['file'])).'<br />';
```

file information

```
// print file last modification time
echo 'File last modified on: '
     .date('Y-m-d', filemtime($_POST['file'])).'<br />';
// is it a directory?
if (is_dir($_POST['file'])) {
   echo 'File is a directory <br />';
}
// is it a file?
if (is_file($_POST['file'])) {
   echo 'File is a regular file <br />';
}
// is it a link?
if (is_link($_POST['file'])) {
   echo 'File is a symbolic link <br />';
}
```

file information

```
      // is it executable?
      if (is_executable($_POST['file'])) {
         echo 'File is executable <br />';
      }
      // is it readable?
      if (is_readable($_POST['file'])) {
         echo 'File is readable <br />';
      }
      // is it writable?
      if (is_writable($_POST['file'])) {
         echo 'File is writable <br />';
      }
   }
   else {
       echo 'File does not exist! <br />';
   }
}
```

file information

```
 ?>
</body>
</html>
```

Agenda

1. Access File
2. Access MySql
3. Access Sqlite

MySql Introduction

- mysql-server, phpMyAdmin
- mysql -u username -p [password]
- CREATE USER 'newuser'@'localhost' IDENTIFIED BY 'password';
- GRANT ALL PRIVILEGES ON * . * TO 'newuser'@'localhost';
- CREATE DATABASE newdbname;
- USE newdbname;
- SHOW DATABASES; SELECT DATABASE();
- SHOW TABLES; DESCRIBE tabname;
- `source` sqlfilename
- \. sqlfilename

Create Table

```
CREATE TABLE symbols (
   id int(11) NOT NULL auto_increment,
   country varchar(255) NOT NULL default '',
   animal varchar(255) NOT NULL default '',
   PRIMARY KEY (id)
);
```

Insert Value

```
INSERT INTO symbols VALUES (1, 'America', 'eagle');
INSERT INTO symbols VALUES (2, 'China', 'dragon');
INSERT INTO symbols VALUES (3, 'England', 'lion');
INSERT INTO symbols VALUES (4, 'India', 'tiger');
INSERT INTO symbols VALUES (5, 'Australia', 'kangaroo');
INSERT INTO symbols VALUES (6, 'Norway', 'elk');
```

Hello, Database

```
<html>
<head>
<basefont face="Arial">
</head>

<body>
<?php
   // set database server access variables:
   $host = "127.0.0.1";
   $user = "root";
   $pass = "xxxxxx";
   $db = "testdb";

   // open connection
   $connection = mysqli_connect($host, $user, $pass)
                  or die ("Unable to connect!");
```

Hello, Database

```
// select database
mysqli_select_db($connection, $db)
   or die ("Unable to select database!");
// create query
$query = "SELECT * FROM symbols";
// execute query
$result = mysqli_query($connection, $query)
  or die ("Error in query: $query. ".mysqli_error($connection));
// see if any rows were returned
if (mysqli_num_rows($result) > 0) {
   // yes
   // print them one after another
   echo "<table cellpadding=10 border=1>";
   while($row = mysqli_fetch_row($result)) {
      echo "<tr>";
      echo "<td>".$row[0]."</td>";
      echo "<td>".$row[1]."</td>";
      echo "<td>".$row[2]."</td>";
```

Hello, Database

```
         echo "</tr>";
      }
      echo "</table>";
   }
   else {
      // no
      // print status message
      echo "No rows found!";
   }
   // free result set memory
   mysqli_free_result($result);

   // close connection
   mysqli_close($connection);
 ?>
</body>
</html>
```

Hello, Database

```
while(list($id, $country, $animal)
      = mysqli_fetch_row($result)) {
   echo "<tr>";
   echo "<td>$id</td>";
   echo "<td>$country</td>";
   echo "<td>$animal</td>";
   echo "</tr>";
}
```

Hello, Database

```
while($row = mysqli_fetch_assoc($result)) {
   echo "<tr>";
   echo "<td>".$row['id']."</td>";
   echo "<td>".$row['country']."</td>";
   echo "<td>".$row['animal']."</td>";
   echo "</tr>";
}
```

Hello, Database

```
while($row = mysqli_fetch_object($result)) {
   echo "<tr>";
   echo "<td>".$row->id."</td>";
   echo "<td>".$row->country."</td>";
   echo "<td>".$row->animal."</td>";
   echo "</tr>";
}
```

OO Access

```
<html>
<head>
<basefont face="Arial">
</head>

<body>
<?php
   // set server access variables
   $host = "127.0.0.1";
   $user = "test";
   $pass = "test";
   $db = "testdb";

   // create mysqli object
   // open connection
   $mysqli = new mysqli($host, $user, $pass, $db);
```

OO Access

```
// check for connection errors
if (mysqli_connect_errno()) {
   die("Unable to connect!");
}

// create query
$query = "SELECT * FROM symbols";
// execute query
if ($result = $mysqli->query($query)) {
   // see if any rows were returned
   if ($result->num_rows > 0) {
      // yes
      // print them one after another
      echo "<table cellpadding=10 border=1>";
      while($row = $result->fetch_array()) {
         echo "<tr>";
         echo "<td>".$row[0]."</td>";
         echo "<td>".$row[1]."</td>";
```

OO Access

```
         echo "<td>".$row[2]."</td>";
         echo "</tr>";
      }
      echo "</table>";
   }
   else {
      // no
      // print status message
      echo "No rows found!";
   }
   // free result set memory
   $result->close();
}
else {
  // print error message
  echo "Error in query: $query. ".$mysqli->error;
}
```

OO Access

```
   // close connection
   $mysqli->close();
 ?>
</body>
</html>
```

PDO

PHP Data Objects, PDO

- Database Access Abstraction Layer
- Reusability (unified API to access multitude of databases, from SQLite to Oracle)
- Cannot perform any database functions using the PDO extension by itself; must use a database-specific PDO driver to access a database server

PDO

```
$host = '127.0.0.1';
$db   = 'test';
$user = 'root';
$pass = '';
$charset = 'utf8';

$dsn = "mysql:host=$host;dbname=$db;charset=$charset";
$opt = [
    PDO::ATTR_ERRMODE            => PDO::ERRMODE_EXCEPTION,
    PDO::ATTR_DEFAULT_FETCH_MODE => PDO::FETCH_ASSOC,
    PDO::ATTR_EMULATE_PREPARES   => false,
];
$pdo = new PDO($dsn, $user, $pass, $opt);
```

PDO

Prepared statements. Protection from SQL injections

```
$stmt = $pdo->prepare('SELECT * FROM users
         WHERE email = ? AND status=?');
$stmt->execute([$email, $status]);
$user = $stmt->fetch();
// or
$stmt = $pdo->prepare('SELECT * FROM users
         WHERE email = :email AND status=:status');
$stmt->execute(['email' => $email, 'status' => $status]);
$user = $stmt->fetch();
```

PDO

```
$sql = "UPDATE users SET name = ? WHERE id = ?";
$pdo->prepare($sql)->execute([$name, $id]);

$stmt = $pdo->prepare("DELETE FROM goods WHERE category = ?");
$stmt->execute([$cat]);
$deleted = $stmt->rowCount();
```

Agenda

1. Access File

2. Access MySql

3. Access Sqlite

Sqlite Introduction

sqlite, sqlite3

```
sqlite3 library.db
>.open library.db
>.databases # show the db file path
>create table books (
     id integer primary key,
     title varchar(255) not null,
     author varchar(255) not null );
>.tables
>.schema tabname
>select * from books;
>.read sqlfilename
>.exit
```

Sqlite Introduction

```
> insert into books (title, author)
  values ('The Lord Of The Rings', 'J.R.R. Tolkien');
> insert into books (title, author)
  values ('The Murders In The Rue Morgue', 'Edgar Allen Poe');
> insert into books (title, author)
  values ('Three Men In A Boat', 'Jerome K. Jerome');
> insert into books (title, author)
  values ('A Study In Scarlet', 'Arthur Conan Doyle');
> insert into books (title, author)
  values ('Alice In Wonderland', 'Lewis Carroll');
```

Sqlite Access

```
<html>
<head></head>

<body>
<?php
   // set path of database file
   $db = $_SERVER['DOCUMENT_ROOT']."/../library.db";
   // open database file
   $handle = sqlite3_open($db)
      or die("Could not open database");
   // generate query string
   $query = "SELECT * FROM books";
   // execute query
   $result = sqlite3_query($handle, $query)
      or die("Error in query: "
          .sqlite3_error_string(sqlite3_last_error($handle)));
```

Sqlite Access

```
// if rows exist
if (sqlite3_num_rows($result) > 0) {
   // get each row as an array
   // print values
   echo "<table cellpadding=10 border=1>";
   while($row = sqlite3_fetch_array($result)) {
      echo "<tr>";
      echo "<td>".$row[0]."</td>";
      echo "<td>".$row[1]."</td>";
      echo "<td>".$row[2]."</td>";
      echo "</tr>";
   }
   echo "</table>";
}
```

Sqlite Access

```
   // all done
   // close database file
   sqlite3_close($handle);
 ?>
</body>
</html>
```

Sqlite Access

```
while($obj = sqlite3_fetch_object($result)) {
   echo "<tr>";
   echo "<td>".$obj->id."</td>";
   echo "<td>".$obj->title."</td>";
   echo "<td>".$obj->author."</td>";
   echo "</tr>";
}
```

Sqlite Access

```
// get the complete result set as a series of nested arrays
$data = sqlite3_fetch_all($result);
// all done close database file
sqlite3_close($handle);
// check the array to see if it contains at least one record
if (sizeof($data) > 0) {
   echo "<table cellpadding=10 border=1>";
   // iterate over outer array (rows)
   // print values for each element of inner array (columns)
   foreach ($data as $row) {
      echo "<tr>";
      echo "<td>".$row[0]."</td>";
      echo "<td>".$row[1]."</td>";
      echo "<td>".$row[2]."</td>";
      echo "</tr>";
   }
   echo "</table>";
}
```

Sqlite Access

```
if (sqlite3_num_rows($result) > 0) {
   echo "<table cellpadding=10 border=1>";
   // check for more rows
   while (sqlite3_has_more($result)) {
      // get first field from each row
      // print values
      $row = sqlite3_fetch_single($result);
      echo "<tr>";
      echo "<td>".$row."</td>";
      echo "</tr>";
   }
   echo "</table>";
}
```

Object-Oriented SQLite API

```
<html>
<head></head>

<body>
<?php
   // set path of database file
   $file = $_SERVER['DOCUMENT_ROOT']."/../library.db";
   // create database object
   $db = new SQLiteDatabase($file)
          or die("Could not open database");
   // generate query string
   $query = "SELECT *  FROM books";
   // execute query
   // return result object
   $result = $db->query($query) or die("Error in query");
```

Object-Oriented SQLite API

```
// if rows exist
if ($result->numRows() > 0) {
   // get each row as an array
   // print values
   echo "<table cellpadding=10 border=1>";
   while($row = $result->fetch()) {
      echo "<tr>";
      echo "<td>".$row[0]."</td>";
      echo "<td>".$row[1]."</td>";
      echo "<td>".$row[2]."</td>";
      echo "</tr>";
   }
   echo "</table>";
}
```

Object-Oriented SQLite API

```
   // all done
   // destroy database object
   unset($db);
 ?>
</body>
</html>
```

Object-Oriented SQLite API

```
if ($result->numRows() > 0) {
   echo "<table cellpadding=10 border=1>";
   // check for more rows
   while ($result->valid()) {
      // get first field from each row
      // print values
      $row = $result->fetchSingle();
      echo "<tr>";
      echo "<td>".$row."</td>";
      echo "</tr>";
   }
   echo "</table>";
}
```

Thank You!

Any Questions?