

Setting up a web site with web server, PHP and MySQL on Windows – by USBWebserver

By Tam Wing Ching, 4/2022

Introduction

Web server is the program which can respond the web page requests (http/https) by sending back the corresponding web page files. You can set up a server at home to let people visit your web pages.

PHP is a programming language used usually on web servers. It is very suitable for doing programming tasks on web pages.

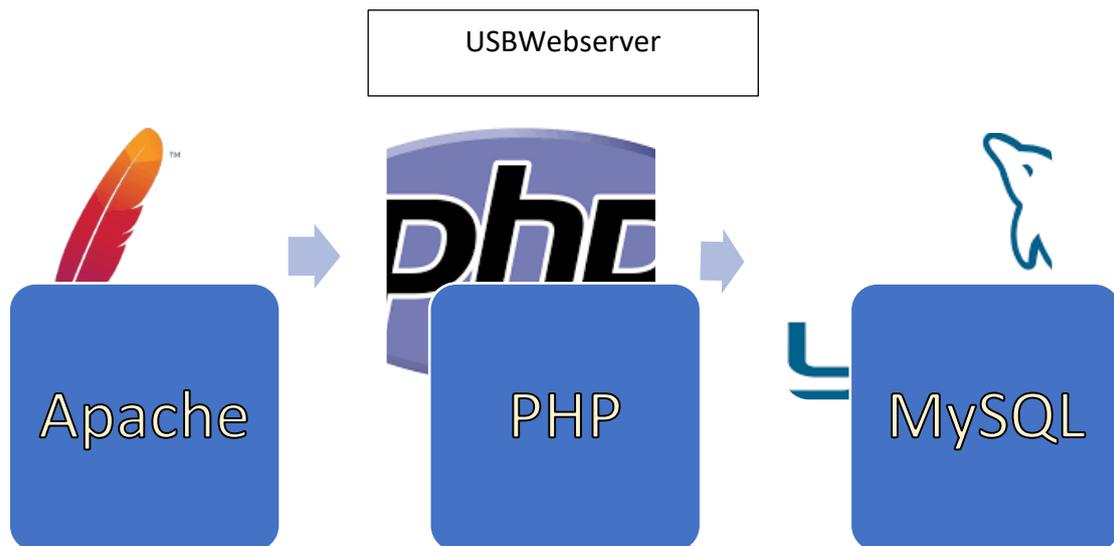
MySQL is a DataBase Management System (DBMS). It is for managing database tables by storing data and collecting data effectively through a language called Structural Query Language (SQL).

In order to set up a web site with data managements, the above three software is a perfect team.

How to Install USB webserver

If you want to install Apache (web server), PHP and MySQL on your computer to test on application systems on web pages, they are usually installed separately.

USBwebserver is a single “non-install” program package to help you to do the above tasks in a batch, saving you a lot of efforts.



It could be downloaded from <https://www.usbwebserver.net/webserver/>, go to “Download USBWebserver V8”. A zip file will be downloaded. Then unzip the zip file and save all the contents inside into a folder of an USB flash drive. Now you can use the USB drive at any Windows systems. The files are listed below.

>	apache2	24 Sept 2013 at 8:17 PM
>	lang	24 Sept 2013 at 8:17 PM
>	mysql	24 Sept 2013 at 8:17 PM
>	php	24 Sept 2013 at 8:18 PM
>	phpmyadmin	24 Sept 2013 at 8:17 PM
>	root	24 Sept 2013 at 8:17 PM
>	settings	24 Sept 2013 at 8:17 PM
	usbwebserver.exe	30 Jul 2013 at 3:38 PM

Run the “usbserver.exe” file on a Windows system.

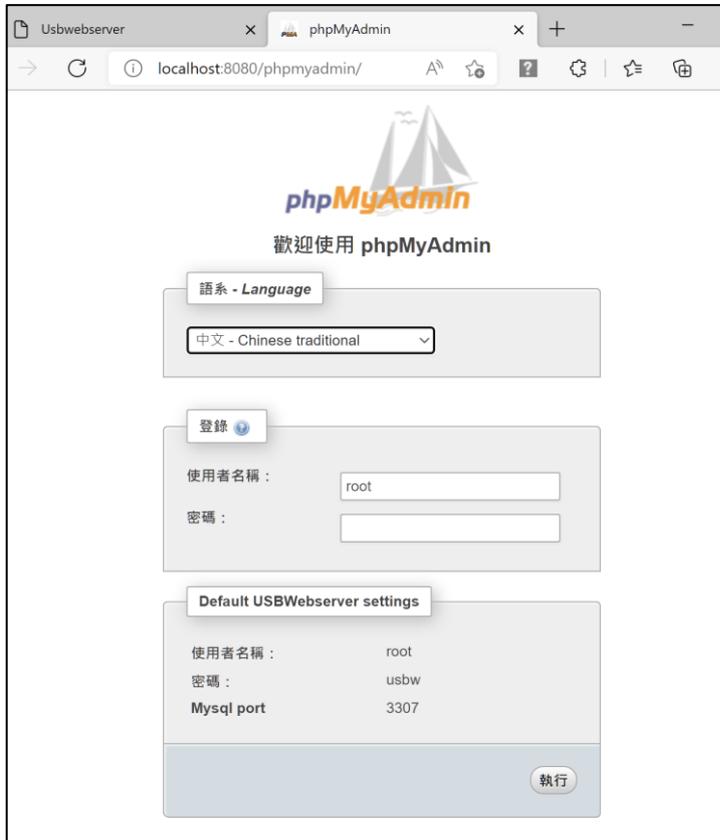


Some explanations

Root dir – launch file explorer to open the root folder of the server.

Localhost – launch web browser to visit the index home page of your web server. It is the same as using a web browser and type in “127.0.0.1:8080” or “localhost:8080”.

PHPMYAdmin – launch web browser to visit the administration page of the PHP admin. . It is the same as using a web browser and type in “localhost:8080/phpmyadmin”.



In order to use the web server, write your home page and save it as “index.php” and save the file in the USBWebserver root folder.

images	7/31/2013 12:20 AM
index.php	7/31/2013 12:18 AM
style.css	7/31/2013 12:17 AM

You can put in:

```
<html>
<body>
<h1> Hello World </h1>

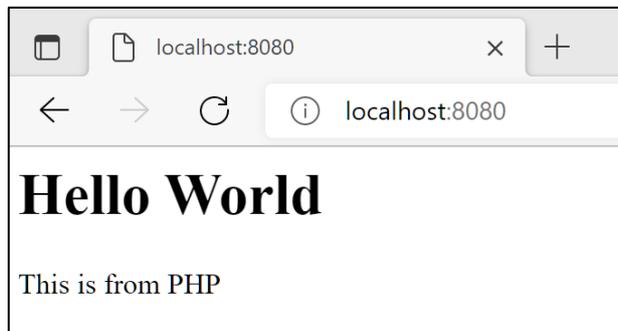
<?php
echo ( "This is from PHP " ) ;
?>

</body>
</html>
```

Save your file as “index.php” and replace the existing “index.php” file.

“Hello World” is the display result of web browser upon receiving the web page HTML.

“This is from PHP” is the execution result of PHP.



If you can see the above, your USBWebserver on Apache and PHP are working fine.

For more information about HTML, you can visit on W3C school -

<https://www.w3schools.com/html/default.asp>

Or a video

<https://www.w3schools.com/videos/index.php>

MySQL server

It is a DBMS server, which enables users to create and search database files. To access it, on a browser, type in “localhost:8080/phpMyAdmin”

Then put in the password “usbw”. (You can change it afterwards in the file “my.ini”, password section if you know more.)

localhost:8080/phpmyadmin/

phpMyAdmin

歡迎使用 phpMyAdmin

語系 - Language

中文 - Chinese traditional

登錄

使用者名稱 : root

密碼 : usbw

Default USBWebserver settings

使用者名稱 :	root
密碼 :	usbw
Mysql port	3307

執行

After log in, click on “SQL” on menu bar.

localhost:8080/phpmyadmin/

localhost

Databases SQL Status Users Export Import More

Run SQL query/queries on server "localhost":

1

Clear

[Delimitter ;] Show this query here again Retain query box Go

Paste the following SQL onto it and press “Go” afterwards.

```
CREATE TABLE IF NOT EXISTS `user` (  
  `uid` varchar(20) NOT NULL,  
  `email` varchar(255) DEFAULT NULL,  
  `pwd` varchar(100) NOT NULL,  
  PRIMARY KEY (`uid`)  
);
```

By default, there is a database called “test”.

After executing the above, it tells MySQL to create a table called “user” under “test” and it defines 3 fields: “uid”, “email”, “pwd”. In which “uid” is the primary key field.

Then paste the following onto SQL and press “Go”.

```
INSERT INTO `user` (`uid`, `email`, `pwd`) VALUES  
( 'user1', 'user1@gmail.com', 'pass1'),  
( 'user2', 'user2@gmail.com', 'pass2'),  
( 'user3', 'user3@gmail.com', 'pass3');
```

It inserts 3 records into table “user”, each with 3 fields as shown below.

				uid	email	pwd
<input type="checkbox"/>	 Edit	 Copy	 Delete	user1	user1@gmail.com	pass1
<input type="checkbox"/>	 Edit	 Copy	 Delete	user2	user2@gmail.com	pass2
<input type="checkbox"/>	 Edit	 Copy	 Delete	user3	user3@gmail.com	pass3
	<input type="checkbox"/> Check All	With selected:	 Change	 Delete	 Export	

For more information about SQL, you can visit on W3C school - <https://www.w3schools.com/sql/default.asp>

Connection with MySQL in PHP

Paste the following codes into "index.php". Use a browser to visit: <http://localhost:8080>

```
<html>
<body>

<form action="" method="POST">
<div id="login">

  <div id="loginbar">Test Login</div>
  <div id="loginmain">
    <table class="noborder">
      <tr>
        <td>Username:</td><td><input type="text" name="uid" size="20" /></td>
      </tr>
      <tr>
        <td>Password:</td><td><input type="password" name="pwd" size="20"
/></td>
      </tr>
    </table>

    <input type="submit" name="btnSubmit" value="Login"><br />

  </div>
</div>
</form>

<?php //this is the program to check if there is a match with the inputted username
and password has a record inside "users" table. If no, echo a line "Login Not
Sucessful!" else echo a line "Login Sucessful!"
$dbServer = "localhost"; //connect to local database server, i.e. MySQL
$dbUser = "root"; //the default username
$dbPass = "usbw"; //the default password
$dbName = "test"; //the default database

$con = mysqli_connect($dbServer, $dbUser, $dbPass, $dbName); //establish
connection
if (!$con){ //if returned no connection established
    die("Could not connect database:". mysqli_error()); //give a warning if failed
}

if (isset($_REQUEST["btnSubmit"])) { //when btnSubmit form is submitted
    $username=$_REQUEST['uid']; //the content of input box uid
    $password=$_REQUEST['pwd']; //the content of input box pwd
    $rs=mysqli_query($con, "select email, pwd from users"); //$rs is the result set of
the SQL search records in table users

    $i = 0;
    while($row=mysqli_fetch_assoc($rs)) //for each record $row
```

```

    {
        $email=$row['email']; //email field
        $pwd=$row['pwd']; //pwd field
        if (($username == $email) && ($password == $pwd)) {
            $i = 1; //denote found username and password
        }
    }
    if (!$i) {
        echo ('Login Not Successful!');
    }
    else echo ('Login Successful!');
}
?>
</body>
</html>

```

Type in username and password in the two input boxes. Then press “Login” button. Still remember the table “users”? We have put in 3 records inside. Use “email” and “pwd” in the same record as the username and password in the two input boxes. E.g. Username: **user1.gmail.com**, Password: **pass1**

uid	email	pwd
user1	user1@gmail.com	pass1
user2	user2@gmail.com	pass2
user3	user3@gmail.com	pass3

Test Login

Username:

Password:

Login Successful!

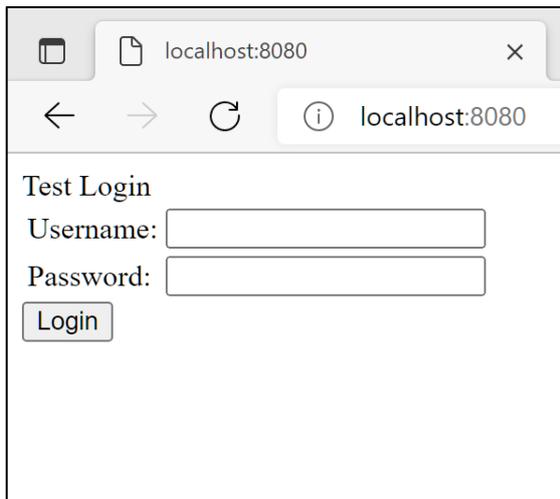
Test Login

Username:

Password:

Login Not Successful!

Program explanations



The screenshot shows a web browser window with the address bar set to localhost:8080. The page content includes the text 'Test Login', followed by a 'Username:' label and a text input field. Below that is a 'Password:' label and a password input field. At the bottom of the form is a button labeled 'Login'.

The above is a form which has two input boxes and a button.

The codes to produce the above form are as follow:

```
<form action="" method="POST">
<div id="login">

  <div id="loginbar">Test Login</div>
  <div id="loginmain">
    <table class="noborder">
      <tr>
        <td>Username:</td><td><input type="text" name="uid" size="20" /></td>
      </tr>
      <tr>
        <td>Password:</td><td><input type="password" name="pwd" size="20"
/></td>
      </tr>
    </table>

    <input type="submit" name="btnSubmit" value="Login"><br />

  </div>
</div>
</form>
```

Please pay attention at the id names:

- "uid" – the name of the username input
- "pwd" – the name of the password input
- "btnSubmit" - the name of the form

They are going to be used in the PHP program on below.

For more information about HTML forms:

https://www.w3schools.com/html/html_forms.asp

```

<?php //this is the program to check if there is a match with the inputted username
and password has a record inside "users" table. If no, echo a line "Login Not
Sucessful!" else echo a line "Login Sucessful!"
$dbServer = "localhost"; //connect to local database server, i.e. MySQL
$dbUser = "root"; //the default username
$dbPass = "usbw"; //the default password
$dbName = "test"; //the default database

$con = mysqli_connect($dbServer, $dbUser, $dbPass, $dbName); //establish
connection
if (!$con){ //if returned no connection established
    die("Could not connect database:". mysql_error()); //give a warning if failed
}

if (isset($_REQUEST["btnSubmit"])) { //when btnSubmit form is submitted
    $username=$_REQUEST['uid']; //the content of input box uid
    $password=$_REQUEST['pwd']; //the content of input box pwd
    $rs=mysqli_query($con, "select email, pwd from users"); //$rs is the result set of
the SQL search records in table users

    $i = 0;
    while($row=mysqli_fetch_assoc($rs)) //for each record $row
    {
        $email=$row['email']; //email field
        $pwd=$row['pwd']; //pwd field
        if (($username == $email) && ($password == $pwd)) {
            $i = 1; //denote found username and password
        }
    }
    if (!$i) {
        echo ('Login Not Successful!');
    }
    else echo ('Login Successful!');
}
?>

```

The functions of the above PHP program:

1. Establish a connection to Table "Test" through MySQL and give warning if failed.
2. Collect the input box contents from the form "btnSubmit" when the "Login" button is pressed.
3. Pass the SQL: "select email, pwd from users" to MySQL. MySQL will return the records of the table "users" on the fields "email" and "pwd".
4. If there is no record with fields matched with the contents of the input boxes ("Username" and "Password"), output "Login Not Successful" else output "Login Sucessful".

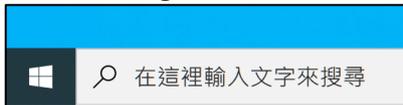
For more information about PHP, you can visit on W3C school -

<https://www.w3schools.com/php/default.asp>

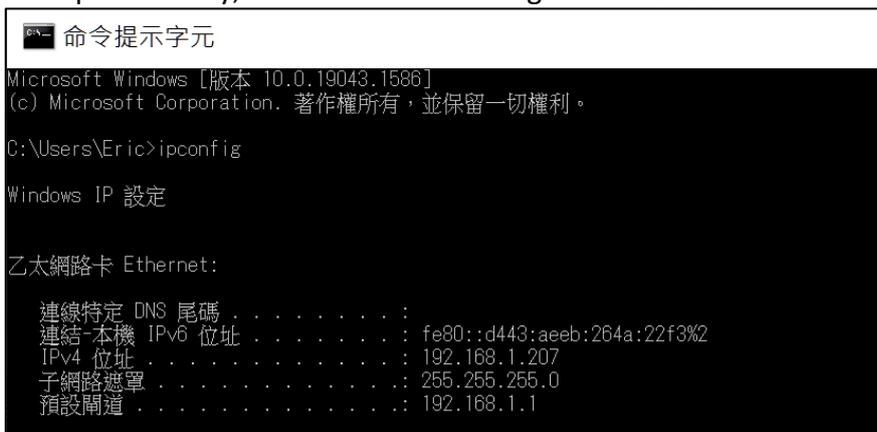
Accessing from WiFi network

To access the USBWebserver server on a mobile phone at home, you must first make both the USBWebserver computer and the mobile joining the same WiFi network.

1. At the USBWebserver computer, type “cmd” at the bottom left corner next to the Windows logo.

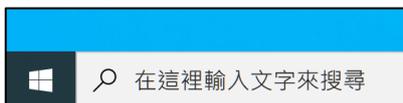


2. Type “ipconfig” to find out the IP address of Windows. It is “192.168.1.207” in the example. Usually, it should be either begin with “192.168” is Class C, local IP address.



(Try Step 5. Below. If it could not work, proceed to Step 3. And Step 4.)

3. Type “Windows Defender” at the bottom left corner to change the settings of Windows Firewall. Turn off temporarily the Firewall for testing purpose. You can set it back on after the test.



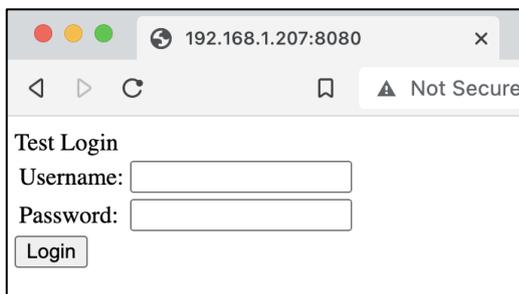


(Optional – may be skipped if only accessed locally at home)

4. Use a browser and type in “192.168.1.1”. It will access the home WiFi router administration page. You must provide the login name and password in order to access the WiFi router. Then you should go to the Network Address Translation (NAT) setting page of your WiFi router and set the port 8080 and the above IP address (“192.168.1.207” in the above example) inside NAT settings.

Since it differs quite a lot in different cases, you must explore by yourself.

5. Now, on a mobile phone or on any other computer in the same WiFi network, use a browser and type in the IP address of the USBWebserver computer (e.g. <http://192.168.1.127:8080>) to access the above login page.



Note: If you look at the source of the login page, it is interesting to find out that the `<?php ?>` part in the “index.php” file disappears, leaving only the html part. It is reasonable because the php part belongs to the server side program and it is not revealed to clients (browsers).

```
192.168.1.207:8080 view-source:192.168.1.207:8080 X +
Not Secure | view-source:192.168.1.207:8080
line wrap
1 <html>
2 <body>
3
4 <form action="" method="POST">
5 <div id="login">
6
7 <div id="loginbar">Test Login</div>
8 <div id="loginmain">
9 <table class="noborder">
10 <tr>
11 <td>Username:</td><td><input type="text" name="uid" size="20" /></td>
12 </tr>
13 <tr>
14 <td>Password:</td><td><input type="password" name="pwd" size="20" /></td>
15 </tr>
16 </table>
17 <input type="submit" name="btnSubmit" value="Login"><br />
18
19 </div>
20 </div>
21 </form>
22
23 </body>
24 </html>
25
26
27
28
```

After testing, you should turn back on the Windows Defender and reset back the NAT settings in WiFi router.