Php Local Server

Introduction

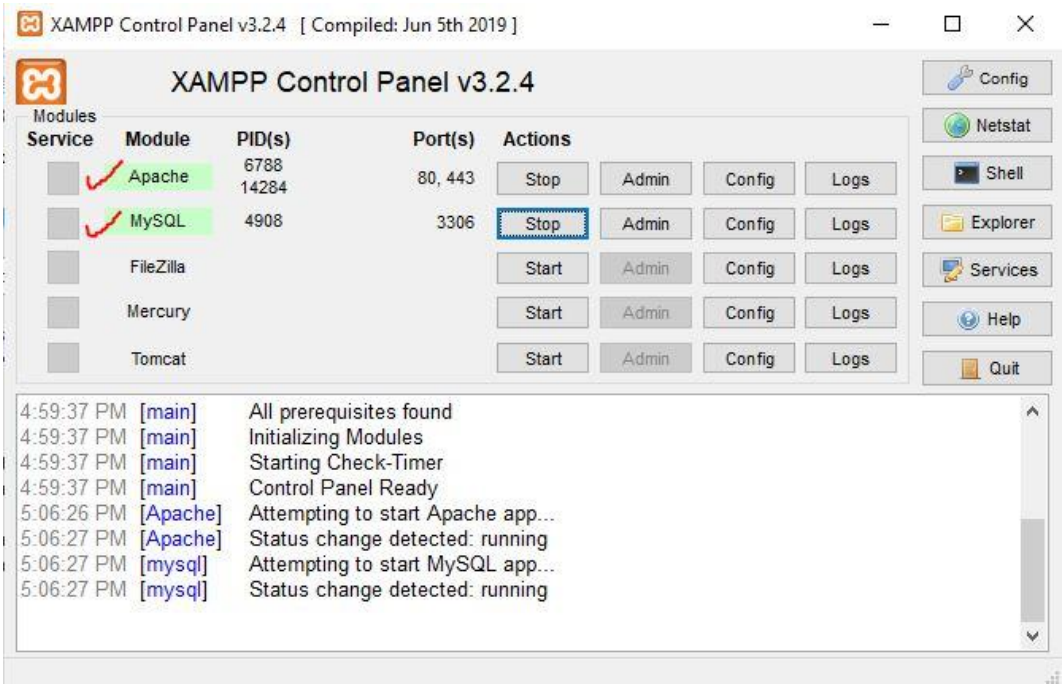
This pill is about to get familiar with php local server, and its critical role in running php projects.

Pill organization

To do this pill we follow the guide line provided in the pill documentation.

1- **Install the XAMPP**

In order to use php local server we need to have php installed on our system. Since in this project we use windows 10 operation system, it is required to install Xampp application. This application is very important, because it make your local system to behave like a server, and client at the same time. Once the Xampp installed, user will have a “Xampp control panel”. We install the Xampp in drive C of the computer. therefor every time we need to run a php file in our browser we need to got this path: c:\xampp and scroll down to find a file called Xampp-controlpanel.exe.



Inside the Xampp control panel we need to click on start button in front of Apache and MySQL. As you can see in the screenshot above.

2- Create project's folder in htdocs

In order to run a file that have php extension (.php) you need to locate your file inside a specific place inside the c drive. In order to do that you need to go through this path → c:\\xampp\htdocs. Inside the htdocs folder you need to create new folder and name it whatever you want. Here I name it "phplocalserver" with no space and all small letter. By doing that you can go to next step.

3- Check you php file in your browser

In order to see your php file inside the browser you need to create a file inside the phplocalserver folder inside the htdocs. We create a file 'main.php'. After you create a php file and type very simple letters inside it (e.g. hello world), then you will save it and open your browser and you need to type 'localhost\phplocalserver\main.php'. by hitting enter you will see the 'hello world' printed in the screen.

4- Check the php version

Php version is important when you are going to run some specific action in your projects. In this project we need to make sure the we have **php version higher than 7.1.3**. in order to check php version, we create a file in our folder with the name main.php. inside this file we wrote this php code.

```
echo phpversion();
```

This line of code will print out the version of our php version on browser. In our case, the current version of our php is 7.4.1.

5- Change time zone to Europe / Madrid

In order to do that we have two options. In this project we do both ways to find out the differences. First option is to change the time in php.ini file. In order to that we open the php.ini file and search for time zone keyword. Infront of time zone we found that it is set to Berlin. By replacing Berlin with Madrid, we can solve this problem.

Second option is to that by code inside our main.php file. Php has a predefined function called date_default_timezone_get() , by using this function and echo it out to the screen you can see that the default time zone is set to Berlin. To change this time zone, we use another predefined function which is job to set new default time zone, inside the parentheses you can put your preferred time zone keyword. In this case we change it to Madrid as follow:

```
date_default_timezone_set("Europe/Madrid");
```

6- Maximize the execution time

Next task is to set the execution time to maximum. In order to do that we need to use to predefined method. To check our current execution time, we use the following code

```
var_dump(ini_get('max_execution_time'));
```

it will return 120 which is default execution time.

in order to change it we use another predefined function of php.

```
ini_set('max_execution_time', 130);
```

this code is used to change the execution time. Few resources claim that if you set it to zero it is going to infinite which is not recommended as best practice.

Now, if we check the previous code again we will see that execution time increased to 130.

7- Enable any extension of choose

In this section we required to enable a disabled extension in php.ini file. In php.ini files there few extensions which is disabled by default.

To enable any extension, you should remove semicolon (**;**) from its beginning.

For this pill we choose GMP extension.

What is GMP?

GMP is a free library for arbitrary precision arithmetic, operating on signed integers, rational numbers, and floating-point numbers. There is no practical limit to the precision except the ones implied by the available memory in the machine GMP runs on. GMP has a rich set of functions, and the functions have a regular interface.

```
Name        : php-gmp
Arch        : x86_64
Version     : 7.2.11
Summary     : A module for PHP applications for using the GNU MP library

Description : These functions allow you to work with arbitrary-length integers using the GNU
MP library.
```

The main target applications for GMP are cryptography applications and research, Internet security applications, algebra systems, computational algebra research, etc.

GMP is carefully designed to be as fast as possible, both for small operands and for huge operands. The speed is achieved by using full words as the basic arithmetic type, by using fast algorithms, with highly optimized assembly code for the most common inner loops for a lot of CPUs, and by a general emphasis on speed.

How to Enable GMP?

As we mentioned by removing the semicolon in front of extension, we can enable it.

8- Add comment to php.ini file

To put comment in php.ini file it is required to place semicolon in front of the line. The line with semicolon in front is ignored in php.ini file.

```
extension=bz2
extension=curl
;extension=ffi
extension=fileinfo
extension=gd2
extension=gettext
extension=gmp  ;uncommented on -22-01-2020
;extension=intl
;extension=imap
;extension=ldap
extension=mbstring
```

As you know, we uncomment GMP extension in php.ini file. Now we are going to write a comment as a hint that we uncomment this extension.