

PHP, Composer tool and Web Server

written by: Hong Le Nguyen, last update 10.24

PHP ---

PHP Configuration File --

C:\Program Files\php8\php.ini

PHP error log config --

```
// C:\Program Files\php8\php.ini file
error_reporting = E_ALL | E_STRICT
log_errors = On
display_errors = Off
error_log = C:/APACHEphp/php_error.log
```

Writing error to log file using error_log()

```
<?php
// Send error message to the server log if error connecting to the database
if (!mysqli_connect("localhost","bad_user","bad_password","my_db")) {
    error_log("Failed to connect to database!", 0);
}
```

PHP infos --

```
// index.php
<?php phpinfo(); ?>
```

PHP build-in web server --

assume that the project directory structure looks like

```
project
  public
    index.php
```

we run PHP build-in web server with document root public/ from project directory

```
$project> php -S localhost:8000 -t public/
```

URL to index.php (page) is

```
http://localhost:8000/
```

run php app in console

```
php public/index.php
```

COMPOSER ---

// install 1 : package install

```
composer require --dev phpunit/phpunit
composer require monolog/monolog
// composer require swiftmailer/swiftmailer
```

result

```
// composer.json
{
  "require-dev": {
    "phpunit/phpunit": "^11.4"
  },
}
```

```

    "require": {
        "monolog/monolog": "^3.8"
    }
}

```

install 2 : packages install

from

```

// composer.json
{
    ...
}

```

install

```
composer install
```

remove a package

```

// remove the PHPUnit library
composer remove phpunit/phpunit

```

autoloader for your classes --

from definition

```

// composer.json
{
    "autoload": {
        "psr-4": { "Hong\\": "src/" }
    },
    ...
}

```

Creating autoloader for your classes in src/

```
composer dump-autoload
```

using autoloader in app --

```

// index.php
require_once __DIR__ . '/vendor/autoload.php';

```

APACHE ---

apache install --

// location

C:\Apache24

apache web server start --

open: terminal for admin

go to: `cd C:\Apache24\bin`

then run

```
C:\Apache24\bin> .\httpd.exe
```

Apache + PHP // Configure PHP as an Apache module --

// C:\Apache24\conf\httpd.conf

open Apache configuration file in a text editor (Notepad).

`cd C:\Apache24\conf`

httpd.conf // file

add the following lines to the bottom of the file to set PHP as an Apache module:

```
# PHP8 module
PHPIniDir "C:/Program Files/php8"
LoadModule php_module "C:/Program Files/php8/php8apache2_4.dll"
AddType application/x-httpd-php .php

<IfModule dir_module>
    DirectoryIndex index.php index.html
</IfModule>
```

APACHE virtual host ---

Step 1: // C:\Windows\System32\drivers\etc\hosts file

```
cd C:\Windows\System32\drivers\etc
open "hosts" file
add the following line to that file and save it as an administrator
127.0.0.1 webstaging.com
```

Step 2: // C:\Apache24\conf\extra\httpd-vhosts.conf file

```
cd C:\Apache24\conf\extra
open "httpd-vhosts.conf" file
add the following code to it
```

```
<VirtualHost *:80>
    DocumentRoot "C:/HONG/PHPtest/webTEST"
    ServerName webstaging.com
    <Directory "C:/HONG/PHPtest/webTEST">
        Options Indexes FollowSymLinks Includes ExecCGI
        AllowOverride All
        Require all granted
    </Directory>
</VirtualHost>
```

=> DOCUMENT ROOT

C:/HONG/PHPtest/webTEST

Step 3: // C:\Apache24\conf\httpd.conf

```
cd C:\Apache24\conf
Open file: httpd.conf
```

enable

```
# Virtual hosts
Include conf/extra/httpd-vhosts.conf
```

browser ---

```
// using virtuel host
http://webstaging.com
```

apacher-web-server-2.4 + PHP-8.3

Links

<https://www.tweakknow.com/RegTweakInstallPHP.php>

<https://dev.to/imradhe/how-to-add-a-custom-virtual-host-on-windows-11-51ai>

<https://njirumwai.hashnode.dev/how-to-setup-apache-virtual-host-on-windows-using-xampp>

Using symfony/console --
run application as console line by using symfony/console component

```
// install
composer require symfony/console

// bin/sym7console file

#!/usr/bin/env php
<?php
require __DIR__.'../vendor/autoload.php';
use Symfony\Component\Console\Application;
use Hong\Command\HelloCommand;

$app = new Application();

// add command to app
$app->add(new HelloCommand());
// $app->add(new AnotherCommand());

// start the console application.
$app->run();

// src/Command/HelloCommand.php

<?php
namespace Hong\Command;
use Symfony\Component\Console\Command\Command;
use Symfony\Component\Console\Input\InputInterface;
use Symfony\Component\Console\Output\OutputInterface;
use Symfony\Component\Console\Input\InputArgument;

class HelloCommand extends Command { // Defining the class

    protected function configure() { // Implementing the configure method
        $this
            ->setName('hello')           // defining the command name
            ->setDescription('Prints Hello') // defining the description
            ->setHelp('This command prints a simple greeting.');// defining the help (shown with -h option)

        $this->addArgument('username', InputArgument::REQUIRED, 'The username of the user.');//
    }

    // implementing the execute method
    protected function execute(InputInterface $input, OutputInterface $output): int {

        // retrieve the argument value using getArguments()
        $username = $input->getArgument('username');
        $output->writeln("Hello, " . $username);

        // returning the success status
        return Command::SUCCESS;
    }
}

// run your app in command line
php bin/sym7console hello <username> as argument>
```