

PHP Composer

Simplifying Dependency
Management in PHP Projects



This project is funded by
the European Union



Maharah
Code Your Future

LibyanSpider

What is Composer?

Composer is a dependency management tool for PHP that allows developers to declare and manage the libraries and packages their projects depend on. It simplifies the process of installing, updating, and autoloading external libraries, making it easier to manage project dependencies and ensure compatibility.



Installing Composer

To start using Composer, follow these steps to install it:

1. Ensure you have PHP installed on your system.
2. Visit the official Composer website at getcomposer.org.
3. Follow the installation instructions for your operating system.
4. Verify the installation by running `composer --version` in your terminal or command prompt.

Creating a Composer.json File

To define the dependencies for your PHP project, you need to create a `composer.json` file in the root directory of your project. This file specifies the required packages and their versions. Here's an example of a `composer.json` file:

```
{
  "require": {
    "vendor/package": "version"
  }
}
```

Installing Dependencies

Once you have defined the dependencies in your `composer.json` file, you can install them by running **`composer install`** in your project's root directory.

Composer will resolve the dependencies and download the required packages into a `vendor` directory.

Updating Dependencies

As your project evolves, you may need to update the dependencies to newer versions. To update all the dependencies listed in your `composer.json` file, run `composer update` in your project's root directory. Composer will check for newer versions and update the packages accordingly.

Autoloading and Using Dependencies

Composer simplifies the process of autoloading dependencies into your PHP project. By default, Composer generates an autoloader that you can include in your project's `index.php` or bootstrap file. It automatically loads the required classes and files from the installed packages, allowing you to use them in your code.

```
require __DIR__.' /vendor/autoload.php' ;
```

Basic Usage

To add Monolog library to your project.

1. Install the latest version with
`composer require monolog/monolog`
2. Use the library:

```
<?php
// monolog-example.php

use Monolog\Level;
use Monolog\Logger;
use Monolog\Handler\StreamHandler;

require __DIR__.'/vendor/autoload.php';

// create a log channel
$log = new Logger('name');
$log->pushHandler(new StreamHandler(__DIR__.'/example.log',
Level::Warning));

// add records to the log
$log->warning('Foo');
$log->error('Bar');
```


Thank you



This project is funded by
the European Union



Maharah
Code Your Future

LibyanSpider