FREE eBook

LEARNING composer-php

Free unaffiliated eBook created from
Stack Overflow contributors.

#composer-php

Table of Contents

About

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: composer-php

It is an unofficial and free composer-php ebook created for educational purposes. All the content is extracted from Stack Overflow Documentation, which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official composer-php.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Chapter 1: Getting started with composer-php

Remarks

This section provides an overview of what composer-php is, and why a developer might want to use it.

It should also mention any large subjects within composer-php, and link out to the related topics. Since the Documentation for composer-php is new, you may need to create initial versions of those related topics.

Examples

Overview

Composer is a tool for dependency management in PHP. It allows you to declare the libraries your project depends on and it will manage (install/update) them for you.

Composer is not a package manager in the same sense as Yum or Apt are. Yes, it deals with "packages" or libraries, but it manages them on a per-project basis, installing them in a directory (e.g. vendor) inside your project.

Composer requires PHP 5.3.2+ to run. A few sensitive php settings and compile flags are also required, but when using the installer you will be warned about any incompatibilities.

To install packages from sources instead of simple zip archives, you will need git, svn, fossil or hg depending on how the package is version-controlled.

Installing Composer on Ubuntu

Before we download and install Composer, we need to make sure our server has all dependencies installed.

First, update the package manager cache by running:

```
sudo apt-get update
```

Now, let's install the dependencies. We'll need `curl` in order to download Composer and `php5-cli` for installing and running it. `git` is used by Composer for downloading project dependencies. Everything can be installed with the following command:

```
sudo apt-get install curl php5-cli git
```

Now let's install it:

```
curl -sS https://getcomposer.org/installer | sudo php -- --install-dir=/usr/local/bin --
filename=composer
```

This will download and install Composer as a system-wide command named composer, under /usr/local/bin. The output should look like this:

```
Output
#!/usr/bin/env php
All settings correct for using Composer
Downloading...

Composer successfully installed to: /usr/local/bin/composer
Use it: php /usr/local/bin/composer
```

To test your installation, run:

```
composer
```

And you should get output similar to this:

```
Output
   ______
  / ____/___  ____ ___  ____  ____  ________  _____
 / /   / __ \/ __ `__ \/ __ \/ __ \/ ___/ _ \/ ___/
/ /___/ /_/ / / / / / / /_/ / /_/ (__  )  __/ /
\____/\____/_/ /_/ /_/ .___/\____/____/\___/_/
                    /_/
Composer version 1.0-dev (9859859f1082d94e546aa75746867df127aa0d9e) 2015-08-17 14:57:00

Usage:
 command [options] [arguments]

Options:
 --help (-h)           Display this help message
 --quiet (-q)          Do not output any message
 --verbose (-v|vv|vvv) Increase the verbosity of messages: 1 for normal output, 2 for more
verbose output and 3 for debug
 --version (-V)        Display this application version
 --ansi                Force ANSI output
 --no-ansi             Disable ANSI output
 --no-interaction (-n) Do not ask any interactive question
 --profile             Display timing and memory usage information
 --working-dir (-d)    If specified, use the given directory as working directory.

....
```

Installing on Windows

Here we will simply use the installer.

This is the easiest way to get Composer set up on your machine.

Download and run Composer-Setup.exe. It will install the latest composer version and set up your `PATH` so that you can just call `composer` from any directory in your command line.

Note: Close your current terminal. Test usage with a new terminal: This is important since the PATH only gets loaded when the terminal starts.

Note-2: Set up `PATH` in windows 10

1. Right click on start up(windows logo)->`system ->Advance system settings->Environment variables->System variables[below box] ->`select `Path` and click `Edit`
2. Click New and add this value `C:\ProgramData\ComposerSetup\bin`
3. Now open your terminal [cmd] and test `composer --version`

Read Getting started with composer-php online: https://riptutorial.com/composer-php/topic/3267/getting-started-with-composer-php

Chapter 2: Auto loading with composer

Examples

Autoloading

For libraries that specify autoload information, Composer generates a vendor/autoload.php file. You can simply include this file and you will get autoloading for free.

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

This makes it really easy to use third party code. For example: If your project depends on Monolog, you can just start using classes from it, and they will be autoloaded.

```
$log = new Monolog\Logger('name');
$log->pushHandler(new Monolog\Handler\StreamHandler('app.log', Monolog\Logger::WARNING));
$log->addWarning('Foo');
```

You can even add your own code to the autoloader by adding an `autoload` field to `composer.json`

```
{
    "autoload": {
        "psr-4": {"Acme\\": "src/"}
    }
}
```

Composer will register a PSR-4 autoloader for the Acme namespace.

You define a mapping from namespaces to directories. The `src` directory would be in your project root, on the same level as vendor directory is. An example filename would be `src/Foo.php` containing an `Acme\Foo` class.

After adding the autoload field, you have to re-run `dump-autoload` to re-generate the `vendor/autoload.php` file.

Including that file will also return the autoloader instance, so you can store the return value of the include call in a variable and add more namespaces. This can be useful for autoloading classes in a test suite, for example.

```
$loader = require __DIR__ . '/vendor/autoload.php';
$loader->add('Acme\\Test\\', __DIR__);
```

In addition to PSR-4 autoloading, Composer also supports PSR-0, classmap and files autoloading.

Read Auto loading with composer online: https://riptutorial.com/composer-php/topic/5915/auto-loading-with-composer

Chapter 3: How to use private repositories with Composer

Parameters

Parameters	Details
repositories	Tells Composer where it can download the required packages.
type: vcs	Tells Composer how to treat the repository.
url: http://...	Tells Composer where is the repository.

Remarks

Use the `type: "vcs"` syntax to use private repositories.

To manage access to the private repository while developing on a local machine, use an `auth.json` file and don't commit it in you project repository. Instead, give access to each single developer to the private repository so, using each one his/her own NOT COMMITTED `auth.json` file, they can fetch the remote repository with `composer install` or `composer update`.

Tip: Put the `auth.json` file in the `.gitignore` file of your `git` repository.

If you are using a continuous integration system, use the `COMPOSER_AUTH` environment variable.

Examples

composer.json syntax

```
{
    "name": "your/package",
    "license": "proprietary",
    "type": "project",
    "description": "How to load an external private Composer package.",
    ...
    "require": {
        "your/private_package": "*"
    },
    ...
    "repositories": [
        {
            "type": "vcs",
            "url": "https://example.com/Your/private-package.git"
        }
    ]
}
```

Read How to use private repositories with Composer online: https://riptutorial.com/composer-php/topic/3554/how-to-use-private-repositories-with-composer

Credits

S. No	Chapters	Contributors
1	Getting started with composer-php	Abdelaziz Dabebi, Atiqur, Community, Nic Wortel
2	Auto loading with composer	Adil Abbasi
3	How to use private repositories with Composer	Aerendir