



# PHP Composer

Dan McDonald, PhD  
Information Systems and Technology  
Utah Valley University



- Composer
  - PHP History
  - What is Composer?
  - What does Composer do?
  - Installing Composer
- Basic Usage
  - Require
  - Composer.json and Composer.lock
  - Packagist

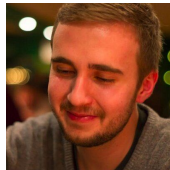


- July 2004 - PHP 5.0
  - Zend engine 2.0
  - PHP 5.5 supported 64-bit builds
  - PEAR (PHP Extension and Application Repository)
    - Repository of PHP software code
    - Used CPAN (Comprehensive Perl Archive Network) as its model
  - PECL (PHP Extension Community Library)
    - Very similar to PEAR and installed with the PEAR package manager
    - Contains C extensions that run more efficiently
- Mar 2012 - Composer
- Dec 2015 - PHP 7.0

# What is Composer?



- Composer is an application-level package manager for the PHP programming language
- Provides a standard format for managing dependencies of PHP software and required libraries
- Created by Nils Adermann, Jordi Boggiano, and others back in 2011



- Strongly inspired by NodeJS's **npm** and Ruby's **bundler**
- Composer is **NOT** like Yum and Apt, which are global by default



# What does Composer do?



- Composer runs through the command line and installs dependencies (libraries) for an application
- Allows users to install PHP applications found on "Packagist" (Packagist.org)
- Packagist is Composer's main repository containing available packages
- Composer allows autoload capabilities



# Composer Requirements

---



- Composer requires PHP 5.3.2+ to run
- To install packages from sources instead of simple zip archives, you will need git, svn, fossil or hg
- Composer is multi-platform and runs on Windows, Linux, and MacOS

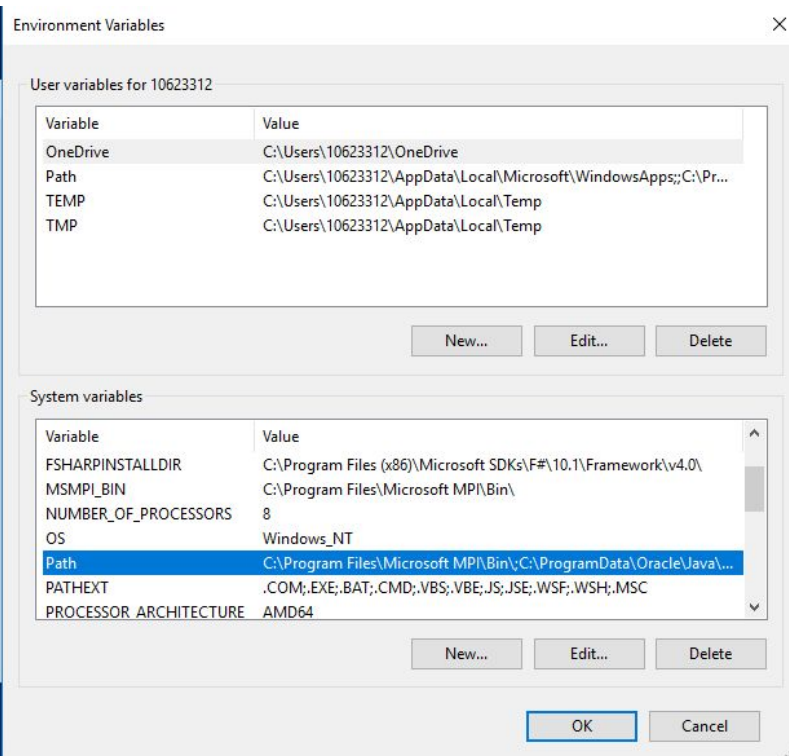
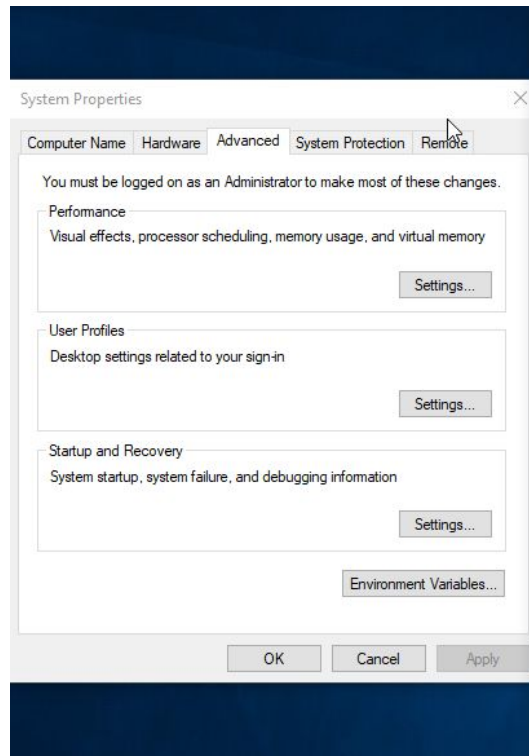


- Go to the page <https://getcomposer.org/download>
- Click on the link to `Composer-Setup.exe` and then launch the installer once it is downloaded
- Just use the defaults in installation, no developer mode, no proxy server
- The current version is v1.8.0



# Adding PHP.exe to the PATH

- Right-click the window button and select System
- type path in search







- To start using Composer in your project, all you need is a `composer.json` file.
- The `composer.json` describes the dependencies of your project and may contain other metadata as well
- The first thing you specify in the `composer.json` is the `require` key
- The default repository is Packagist
- Package Name is a Vendor's name + project's name



## Adding a dependency



- Create a folder called cards in your htdocs directory
- Open the folder using VS Code
- Open the terminal via View>Terminal
- Run command `composer require guzzlehttp/guzzle`
- The require command also runs the install



# Running composer command



PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

1: powershell ▼

```
PS C:\xampp\htdocs\cards> composer require guzzlehttp/guzzle
Using version ^6.3 for guzzlehttp/guzzle
./composer.json has been created
Loading composer repositories with package information
Updating dependencies (including require-dev)
Package operations: 5 installs, 0 updates, 0 removals
- Installing guzzlehttp/promises (v1.3.1): Downloading (100%)
- Installing ralouphie/getallheaders (2.0.5): Downloading (100%)
- Installing psr/http-message (1.0.1): Downloading (100%)
- Installing guzzlehttp/psr7 (1.5.2): Downloading (100%)
- Installing guzzlehttp/guzzle (6.3.3): Downloading (100%)
guzzlehttp/guzzle suggests installing psr/log (Required for using the Log middleware)
Writing lock file
Generating autoload files
```



- The require command creates the `composer.json`
- If no `composer.lock` file exists, one is created with all the current versions of the dependencies listed in your `composer.json` file and downloads the dependencies to your vendor directory
- This file should be committed to repo

```

{ } composer.json x
1  {
2      "require": {
3          "guzzlehttp/guzzle": "^6.3"
4      }
5  }

{ } composer.lock x
1  {
2      "_readme": [
3          "This file locks the dependencies of your project to a
4          "Read more about it at https://getcomposer.org/doc/01-
5          "This file is @generated automatically"
6      ],
7      "content-hash": "55f6c2e4bd979a78cd8c3aa41dfe6691",
8      "packages": [
9          {
10             "name": "guzzlehttp/guzzle",
11             "version": "6.3.3",
12             "source": {
13                 "type": "git",
14                 "url": "https://github.com/guzzle/guzzle.git",
15                 "reference": "407b0cb880ace85c9b63c5f9551db498"
16             },
17             "dist": {
18                 "type": "zip",
19                 "url": "https://api.github.com/repos/guzzle/gu
20                 "reference": "407b0cb880ace85c9b63c5f9551db498"
21                 "shasum": ""
22             },
23             "require": {
24                 "guzzlehttp/promises": "^1.0",
25                 "guzzlehttp/psr7": "^1.4",
26                 "php": ">=5.5"
27             }
28         }
29     ]
30 }
    
```



- If there is already a composer.lock file when you run install, then Composer uses the exact versions listed in the composer.lock file to ensure package versions are consistent for everyone
- You will have all the dependencies requested by your composer.json, but they may not all be the very latest available versions
- Make sure to commit this file to VC, so the correct versions of the dependencies are used



- If you want to update your `composer.lock` file, then run `composer update`
- This will fetch the latest matching version (based on `composer.json`) and update the lock file
- This is equivalent to deleting the `composer.lock` file and running `install` again



# Packagist



- Packagist is the main Composer repository
- You can require any package that is available there
- Any open source project using Composer is recommended to be published there
- A library does not need to be on Packagist to be used by Composer

**GVU**™

---

