**Introducing Dependency Management and Composer**

Links

[Composer](https://getcomposer.org/) - Dependency Manager for PHP

[PHPMailer](https://packagist.org/packages/phpmailer/phpmailer) - full-featured email creation and transfer class for PHP

[OAuth2-Google](https://packagist.org/packages/league/oauth2-google) - Google OAuth 2.0 Client from [The League of Extrodinary Packages](https://thephpleague.com/)

[PHPUnit](https://packagist.org/packages/phpunit/phpunit) - The PHP Unit Testing framework.

**Installing Composer**

#### Tips

1. Install globally
2. Match your development environment
3. Do NOT include your vendor folder in your Version Control System (VCS)

#### Installation - Linux / Unix / OSX

For additional help, see the [Documentation](https://getcomposer.org/doc/00-intro.md#installation-linux-unix-osx)

#### Installation - Windows

For additional help, see the [Documentation](https://getcomposer.org/doc/00-intro.md#installation-windows)

Download and run [Composer-Setup.exe](https://getcomposer.org/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.

Finding and Using Packages

#### Repositories

[Packagist.org](https://packagist.org/) - Default package repository

[Composer Documentation](https://getcomposer.org/doc/05-repositories.md#repository) - The repository chapter of the documentation will explain the concept of packages and repositories, what kinds of repositories are available, and how they work.

#### Packages Used in this Video

##### **CKEditor**

[ckeditor/ckeditor](https://packagist.org/packages/ckeditor/ckeditor) on packagist

What-You-See-Is-What-You-Get (W-Y-S-I-W-Y-G ) editors bring common word processor features directly to your web pages.

[CKEditor](http://ckeditor.com/) is fully [documented](http://docs.ckeditor.com/#!/guide), including a Quick Start Guide for [Adding CKEditor to your page](http://docs.ckeditor.com/#!/guide/dev_installation-section-adding-ckeditor-to-your-page).

##### **HTML Purifier**

[HTML Purifier](http://htmlpurifier.org/docs) is a standards-compliant HTML filter library written in PHP. HTML Purifier will not only remove all malicious code (better known as XSS) with a thoroughly audited, secure yet permissive whitelist, it will also make sure your documents are standards compliant, something only achievable with a comprehensive knowledge of W3C's specifications.

[ezyang/htmlpurifier](https://packagist.org/packages/ezyang/htmlpurifier) on packagist

Managing Package Versions

#### Testing Version Constraints

There is a handy [packagist semver checker](https://semver.mwl.be/" \t "_blank) (short for symantic versioning) that allows you to test out your version constraints.

#### Version Usage [Documentation](https://getcomposer.org/doc/articles/versions.md)

**Exact** - You can specify the exact version of a package. This will tell Composer to install this version and this version only. If other dependencies require a different version, the solver will ultimately fail and abort any install or update procedures.

Maintaining Consistent Environments

Download the Final Project (composer\_complete.zip)

When updating to a patch version, nothing SHOULD break. But if you're trying to track down bugs, you want to make sure that it's not something introduced in someone else's code.

#### Reminder

When updating composer dependencies, make sure you commit your composer.lock file to Version Control system. DO NOT commit your vendor folder.

#### Commands

composer install will look for a composer.lock file first and install the versions specified there. If a lock file does not exist, composer will use the composer.json file to install dependencies.

composer update will update all packages as well as the composer.lock file.

composer show will list all installed packages and their versions