

Composer & Frameworks

How they make development easier



Tobias Derksen

22nd April 2016

Fontys School of Technology & Logistics

Table of contents

1. Frameworks - Do not reinvent the wheel
2. Composer - Dependency Management for PHP
3. Workshop

About me

- 25 years
- Student @ Fontys Venlo
- 2nd Year - Software-Engineering
- <https://github.com/lukeelten>

Frameworks - Do not reinvent the wheel

What is a Framework?

[...] a software framework is an abstraction in which software providing generic functionality [...]

- Wikipedia (Software framework)

What is a Framework?

- Reusable code (DRY)
- Design pattern applied
- Server-side / Client-side

Why should I use a Framework?

- + Faster Coding
- + Security Functions included
- + Tested and frequently used
- + ORM

But ...

- Time to learn
- Forced application structure
- Multiple files to load, Execution time

How to choose

- Company standard
- License
- Closed / Open Source



Contao



Magento®
eCommerce Platform for Growth

PIWIK



shopware



**Massachusetts
Institute of
Technology**





Other important frameworks

- Laravel
- Horde Framework
- Yana Framework
- Yii Framework

Composer - Dependency Management for PHP

What is Dependency Management?

- Centralized dependency management
- Manage dependencies versions
- Automate parts of the build process

Dependency Management in other languages

- Maven (Java)
- gem / bundler (Ruby)
- pip (Python)
- npm / bower (Javascript)

Why composer makes your life easier

- Same versions on different environments
- It is easy to update dependencies
- Composer generates Autoloader
- Only use parts of frameworks

Important files

- `composer.json`
- `composer.lock`
- `vendor/autoload.php`

```
{  
    "name": "vendorName/projectName",  
    "require": {  
        "important/package": "2.*",  
    }  
}
```

Versioning Schemas

Constraint	Meaning
2.0.1	exactly 2.0.1
2.0.*	$\geq 2.0.0 < 2.1.0$
≥ 2.0	
~2.0	$\geq 2.0.0 < 3.0.0$
~2.0.0	$\geq 2.0.0 < 2.1.0$
^2.0.0	$\geq 2.0.0 < 3.0.0$
1.0 - 2.0	$\geq 1.0.0 < 2.1$
$\geq 2.0.*$	illegal

Workshop

Installing composer

OSX

brew install composer

Linux

Package Manager: *composer* or *php-composer*

Download .phar at <https://getcomposer.org/download>

Windows

Installation package on <https://getcomposer.org/download>

Existing PHP Installation required (e.g. XAMPP)

Using composer

General command

composer OR php composer.phar

Create project

composer create-project *vendor/projectname target_dir*

Install package

composer require [--dev] *package_name*

Use existing project

composer install

composer update

Start CakePHP Project

Create Project

```
composer create-project --prefer-dist cakephp/app AppName
```

Thank you for your attention.

Any Questions?

This talk is freely available on GitHub.

<https://github.com/lukeelten/typo3camp2016>

Creative Commons Attribution-ShareAlike 4.0



Sources

- <https://getcomposer.org>
- <http://cakephp.org>
- <http://bakery.cakephp.org>
- https://en.wikipedia.org/wiki/Software_framework
- <http://flow.typo3.org>
- <https://symfony.com>
- <http://framework.zend.com>