



Composer

Présentation

Bien démarrer

Intégration dans  
notre projet

Un peu plus loin

# COMPOSER

## Mon ami pour la vie

Benjamin Lampérier - Benoît Goyheneche

RMLL 2015 - Beauvais

8 Juillet 2015



## Composer

Présentation

Bien démarrer

Intégration dans  
notre projet

Un peu plus loin

- 1 Présentation
- 2 Bien démarrer
- 3 Intégration dans notre projet
- 4 Un peu plus loin



# Introduction

Composer

Présentation

Bien démarrer

Intégration dans  
notre projet

Un peu plus loin

- Composer est un gestionnaire de dépendance open source pour PHP et écrit en PHP.



# Introduction

Composer

Présentation

Bien démarrer

Intégration dans  
notre projet

Un peu plus loin

- Composer est un gestionnaire de dépendance open source pour PHP et écrit en PHP.
- Créé en 2011 par Nils ADERMANN et Jordi BOGGIANO



# Introduction

Composer

Présentation

Bien démarrer

Intégration dans  
notre projet

Un peu plus loin

- Composer est un gestionnaire de dépendance open source pour PHP et écrit en PHP.
- Créé en 2011 par Nils ADERMANN et Jordi BOGGIANO
- Publié sous licence MIT



# Un peu d'histoire

Composer

Présentation

Bien démarrer

Intégration dans  
notre projet

Un peu plus loin

## ■ PHP a 20 ans



# Un peu d'histoire

Composer

Présentation

Bien démarrer

Intégration dans  
notre projet

Un peu plus loin

- PHP a 20 ans
- PHP 7 arrive



# Un peu d'histoire

Composer

Présentation

Bien démarrer

Intégration dans  
notre projet

Un peu plus loin

- PHP a 20 ans
- PHP 7 arrive
- PHP - fig





# Un peu d'histoire

Composer

Présentation

Bien démarrer

Intégration dans  
notre projet

Un peu plus loin

- PHP a 20 ans
- PHP 7 arrive
- PHP - fig
- Toujours plus d'outils pour nous aider



# Des dépendances qui...

Composer

Présentation

Bien démarrer

Intégration dans  
notre projet

Un peu plus loin



# Des dépendances qui...

Composer

Présentation

Bien démarrer

Intégration dans  
notre projet

Un peu plus loin

- **doivent** être mises à jour



# Des dépendances qui...

Composer

Présentation

Bien démarrer

Intégration dans  
notre projet

Un peu plus loin

- **doivent** être mises à jour
- peuvent avoir des dépendances à d'autres bibliothèques



# Des dépendances qui...

Composer

Présentation

Bien démarrer

Intégration dans  
notre projet

Un peu plus loin

- **doivent** être mises à jour
- peuvent avoir des dépendances à d'autres bibliothèques
- ont chacune leurs paramètres d'*autoload*



# Installation

Composer

Présentation

Bien démarrer

Intégration dans  
notre projet

Un peu plus loin

## ■ PHP fonctionnel



# Installation

Composer

Présentation

Bien démarrer

Intégration dans  
notre projet

Un peu plus loin

- PHP fonctionnel
- PHP 5.4.3 minimum



# Installation

Composer

Présentation

Bien démarrer

Intégration dans  
notre projet

Un peu plus loin

- PHP fonctionnel
- PHP 5.4.3 minimum
- curl





# Installation

Composer

Présentation

Bien démarrer

Intégration dans  
notre projet

Un peu plus loin

- PHP fonctionnel
- PHP 5.4.3 minimum
- curl
- `curl -sS https://getcomposer.org/installer | php`

```
CatichedeBenoit:rml1 Benoit$ curl -sS https://getcomposer.org/installer | php
#!/usr/bin/env php
All settings correct for using Composer
Downloading...

Composer successfully installed to: /Users/Benoit/Desktop/rml1/composer.phar
Use it: php composer.phar
CatichedeBenoit:rml1 Benoit$ █
```



# Installation

Composer

Présentation

Bien démarrer

Intégration dans  
notre projet

Un peu plus loin

- on a une installation locale



# Installation

Composer

Présentation

Bien démarrer

Intégration dans  
notre projet

Un peu plus loin

- on a une installation locale
- `mv composer.phar /usr/local/bin/composer`



# Installation

Composer

Présentation

Bien démarrer

Intégration dans  
notre projet

Un peu plus loin

- on a une installation locale
- `mv composer.phar /usr/local/bin/composer`
- Gestion des bibliothèques de votre projet



# Packagist

Composer

Présentation

Bien démarrer

Intégration dans  
notre projet

Un peu plus loin

■ `https://packagist.org/`



# Packagist

Composer

Présentation

Bien démarrer

Intégration dans  
notre projet

Un peu plus loin

- `https://packagist.org/`
- Dépôt des paquets par défaut



# Packagist

Composer

Présentation

Bien démarrer

Intégration dans  
notre projet

Un peu plus loin

- <https://packagist.org/>
- Dépôt des paquets par défaut
- Hook pour gérer les mise à jour



# Fonctionnement

Composer

Présentation

Bien démarrer

Intégration dans  
notre projet

Un peu plus loin

- Composer lit et agit en fonction de ce qu'il trouve dans le fichier `composer.json` à la racine de notre projet





# Fonctionnement

Composer

Présentation

Bien démarrer

Intégration dans  
notre projet

Un peu plus loin

- Composer lit et agit en fonction de ce qu'il trouve dans le fichier `composer.json` à la racine de notre projet
- 2 parties importantes :

```
...  
    "require": {  
        "monolog/monolog": "1.0.*",  
        "joli/ternel": "@dev",  
        "acme/foo": "dev-master#2eb0c0978d290a1c45346a1955188929cb4e5db7"  
    },  
    "require-dev": {  
        "debug/dev-only": "1.0.*"  
    },  
    ...  
}
```



# Bien démarrer

Composer

Présentation

Bien démarrer

Intégration dans  
notre projet

Un peu plus loin

## ■ composer init

```
CatichedeBenoit:rml1 Benoit$ composer init
```

```
Welcome to the Composer config generator
```

This command will guide you through creating your composer.json config.

```
Package name (<vendor>/<name>) [Benoit/rml1]: █
```



# Bien démarrer

Composer

Présentation

Bien démarrer

Intégration dans  
notre projet

Un peu plus loin

## ■ `composer init`

```
CatichedeBenoit:rml1 Benoit$ composer init
```

```
Welcome to the Composer config generator
```

This command will guide you through creating your `composer.json` config.

```
Package name (<vendor>/<name>) [Benoit/rml1]: █
```

- On peut chercher les paquets que l'on souhaite utiliser si on les connaît (`require + dev`)



# Bien démarrer

Composer

Présentation

Bien démarrer

Intégration dans  
notre projet

Un peu plus loin

## ■ `composer init`

```
CatichedeBenoit:rml1 Benoit$ composer init
```

```
Welcome to the Composer config generator
```

```
This command will guide you through creating your composer.json config.
```

```
Package name (<vendor>/<name>) [Benoit/rml1]: █
```

- On peut chercher les paquets que l'on souhaite utiliser si on les connaît (require + dev)
- Sur packagist et on copie la première ligne :  

```
composer require zetacomponents/graph  
composer install
```



# Intégration dans notre projet

Composer

Présentation

Bien démarrer

**Intégration dans  
notre projet**

Un peu plus loin



# Intégration dans notre projet

Composer

Présentation

Bien démarrer

Intégration dans  
notre projet

Un peu plus loin

## ■ Magie de l'autoload



# Intégration dans notre projet

Composer

Présentation

Bien démarrer

Intégration dans  
notre projet

Un peu plus loin

- Magie de l'autoload
- `require '/vendor/autoload.php';`



# Utiliser ses paquets

Composer

Présentation

Bien démarrer

Intégration dans  
notre projet

Un peu plus loin

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





# Utiliser ses paquets

Composer

Présentation

Bien démarrer

Intégration dans  
notre projet

Un peu plus loin

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

■ use \App\SousDossier\MonFichier



# Utiliser ses paquets

Composer

Présentation

Bien démarrer

Intégration dans  
notre projet

Un peu plus loin

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

- `use \App\SousDossier\MonFichier`
- `composer dump-autoload`



# Pour résumer

Composer

Présentation

Bien démarrer

Intégration dans  
notre projet

Un peu plus loin

## ■ Installation de composer



# Pour résumer

Composer

Présentation

Bien démarrer

Intégration dans  
notre projet

Un peu plus loin

- Installation de composer
- `composer init`



# Pour résumer

Composer

Présentation

Bien démarrer

Intégration dans  
notre projet

Un peu plus loin

- Installation de composer
- `composer init`
- `composer require vendor/package`



# Pour résumer

Composer

Présentation

Bien démarrer

Intégration dans  
notre projet

Un peu plus loin

- Installation de composer
- `composer init`
- `composer require vendor/package`
- `composer install`



# Pour résumer

Composer

Présentation

Bien démarrer

Intégration dans  
notre projet

Un peu plus loin

- Installation de composer
- `composer init`
- `composer require vendor/package`
- `composer install`
- Ajout de l'autoloader



# Gestion des versions

Composer

Présentation

Bien démarrer

Intégration dans  
notre projet

Un peu plus loin

- "vendor/name" : "x.y.z"





# Gestion des versions

Composer

Présentation

Bien démarrer

Intégration dans  
notre projet

Un peu plus loin

- "vendor/name" : "x.y.z"
- Une version précise : 0.9.88



# Gestion des versions

Composer

Présentation

Bien démarrer

Intégration dans  
notre projet

Un peu plus loin

- "vendor/name" : "x.y.z"
- Une version précise : 0.9.88
- Une version avec joker : 1.2.\*



# Gestion des versions

Composer

Présentation

Bien démarrer

Intégration dans  
notre projet

Un peu plus loin

- "vendor/name" : "x.y.z"
- Une version précise : 0.9.88
- Une version avec joker : 1.2.\*
- Une plage de version :  $\geq 1.3 < 2.1$



# Gestion des versions

Composer

Présentation

Bien démarrer

Intégration dans  
notre projet

Un peu plus loin

- "vendor/name" : "x.y.z"
- Une version précise : 0.9.88
- Une version avec joker : 1.2.\*
- Une plage de version :  $\geq 1.3 < 2.1$
- Notation avec tiret : 2.0 – 3.0



# Gestion des versions

Composer

Présentation

Bien démarrer

Intégration dans  
notre projet

Un peu plus loin

- "vendor/name" : "x.y.z"
- Une version précise : 0.9.88
- Une version avec joker : 1.2.\*
- Une plage de version :  $\geq 1.3 < 2.1$
- Notation avec tiret : 2.0 – 3.0
- Notation avec tilde :  $\sim 3.2$



# Gestion des versions

Composer

Présentation

Bien démarrer

Intégration dans  
notre projet

Un peu plus loin

- "vendor/name" : "x.y.z"
- Une version précise : 0.9.88
- Une version avec joker : 1.2.\*
- Une plage de version :  $\geq 1.3 < 2.1$
- Notation avec tiret : 2.0 – 3.0
- Notation avec tilde :  $\sim 3.2$
- Notation avec circonflexe :  $\wedge 1.2.3$



# Des commandes bien utiles

Composer

Présentation

Bien démarrer

Intégration dans  
notre projet

Un peu plus loin

- `composer list`



# Des commandes bien utiles

Composer

Présentation

Bien démarrer

Intégration dans  
notre projet

Un peu plus loin

- `composer list`
- `composer remove`





# Des commandes bien utiles

Composer

Présentation

Bien démarrer

Intégration dans  
notre projet

Un peu plus loin

- `composer list`
- `composer remove`
- `composer update`



# Des commandes bien utiles

Composer

Présentation

Bien démarrer

Intégration dans  
notre projet

Un peu plus loin

- `composer list`
- `composer remove`
- `composer update`
- `composer create-project --prefer-dist  
cakephp/app[ app_name ]`



# Des commandes bien utiles

Composer

Présentation

Bien démarrer

Intégration dans  
notre projet

Un peu plus loin

- `composer list`
- `composer remove`
- `composer update`
- `composer create-project --prefer-dist  
cakephp/app[ app_name ]`
- `composer self-update`

# Des questions ?

Composer

Présentation

Bien démarrer

Intégration dans  
notre projet

Un peu plus loin





Composer

Présentation

Bien démarrer

Intégration dans  
notre projet

Un peu plus loin

Merci de votre attention