



# **GIT, COMPOSER ET DRUPAL**

## **GÉRER ET VERSIONNER SON DRUPAL**

# INTRODUCTION

# \$ WHOIS



- Kevin Gautreau
- Développeur Back freelance
- Clermont-Ferrand (depuis Septembre 2015)
- Spécialisé Drupal
- Twitter / Github : @Kgaut
- Mail : [contact@kgaut.net](mailto:contact@kgaut.net)
- Trésorier association Drupal France



**GIT**

# « VERSIONNER »

“ *Mettre sous gestionnaire de version* ”

- Effectuer des "sauvegarde" de son code à un instant T.
- Revenir à une version précédente
- Travailler à plusieurs, fusionner des développements

# WUT ?

Systeme de gestion de version à la mode.

Développé à l'origine par Linus Torvalds.

Démocratisé par Github.

# PRINCIPE SIMPLE

Création d'un dépôt : `git init`

*\*\* Code \*\**

Création d'une version :

```
git commit
```

*\*\* Code \*\**

Création d'une nouvelle version :

```
git commit
```

...

# AJOUT D'UNE "REMOTE"

Pour le travail collaboratif : création d'une copie de notre dépôt.

```
git remote add origin chemin/vers/remote
```

récupération d'un dépôt distant

```
git clone chemin/vers/remote
```

Remotes possible : github, gitlab, dossier local / réseau...



# LE FICHER .GITIGNORE

Fichier à placer à la racine du dépôt

Contenant les dossiers / fichiers à ne pas versionner (images, fichiers de config, css...)

```
db/*.sql  
web/uploads  
web/sites/default/settings.php
```



# COMPOSER

# WUT ?

- Gestionnaire de dépendances PHP
- Permet d'utiliser des « morceaux de code » et de gérer les version / MAJ

# INSTALLER COMPOSER

GNU/Linux & MacOS

```
curl -sS https://getcomposer.org/installer | php  
sudo mv composer.phar /usr/local/bin/composer
```

Windows

<https://kgaut.net/blog/2017/installer-composer-sous-windows.html>

# AJOUTER UNE DÉPENDANCE À NOTRE PROJET

`composer require dependance`

```
composer require silex/silex
```

Va installer silex + ses dépendances

# COMPOSER.JSON

Fichier contenant les dépendances de notre projet  
PHP

```
{  
  "require": {  
    "silex/silex": "^2.2"  
  }  
}
```

# COMPOSER.LOCK

fichier contenant la version exacte de nos dépendances (et de leurs dépendances)

# SUR QUOI SE BASE COMPOSER ?

Un dépôt (repository) : packagist (<https://packagist.org>)

The screenshot shows the Packagist website interface. At the top, there's a navigation bar with 'Packagist The PHP Package Repository' on the left and 'Browse', 'Submit', 'Create account', and 'Sign in' on the right. Below this is a search bar with the placeholder text 'Search packages...'. The main content area displays the package 'silex/silex'. It includes a command to install the package: `composer require silex/silex`. The description reads: 'The PHP micro-framework based on the Symfony Components'. On the right side, there's a profile section for the package maintainer, including a profile picture, a link to 'github.com/silexphp/Silex', and a list of links: 'Homepage', 'Source', and 'Issues'. Below these links is a statistics table:

Installs:	5 514 370
Dependents:	1 660
Suggesters:	46
Stars:	3 653
Watchers:	254
Forks:	739
Open Issues:	37

Below the statistics, there's a section for the current version 'v2.2.0', dated '2017-07-23 07:40 UTC'. This section is divided into three columns: 'requires', 'requires (dev)', and 'suggests'. The 'requires' column lists dependencies like 'php: >=5.5.9', 'pimple/pimple: ~3.0', and various Symfony components. The 'requires (dev)' column lists development dependencies like 'symfony/asset', 'symfony/expression-language', 'symfony/security', etc. The 'suggests' column is currently empty, showing 'None'. On the right side of this section, there's a list of available versions: 'dev-master / 2.2.x-dev', 'v2.2.0' (highlighted), 'v2.1.0', 'v2.0.4', 'v2.0.3', 'v2.0.2', 'v2.0.1', 'v2.0.0', '1.3.x-dev', and 'v1.2.6'.



# CE QUE PERMET COMPOSER

1. De ne pas avoir à télécharger les librairies tierces manuellement
2. De ne pas avoir à *versionner* ces librairies
3. Gérer facilement les maj

# GIT + COMPOSER

1. Versionnement des composer.json et composer.lock
2. Ajout au `.gitignore` du dossier "vendor"
3. `git clone`
4. `composer install`

# MISE À JOUR DE DÉPENDANCES

```
composer update
```

modifiera le fichier `composer.lock` mais pas le `composer.json`Note :

# SPÉCIFIER UNE VERSION PRÉCISE

```
"require": {  
  "silex/silex" "1.3.6"  
}
```

# CONTRAINTE DE VERSION

```
"require": {  
  // >, <, >=, <= | specify upper / lower bounds  
  "vendor/package": ">=1.3.2", // anything above or equal to 1.3.2  
  "vendor/package": "<1.3.2", // anything below 1.3.2  
  
  // * | wildcard  
  "vendor/package": "1.3.*", // >=1.3.0 <1.4.0  
  
  // ~ | allows last digit specified to go up  
  "vendor/package": "~1.3.2", // >=1.3.2 <1.4.0  
  "vendor/package": "~1.3", // >=1.3.0 <2.0.0  
  
  // ^ | doesn't allow breaking changes (major version fixed - following semver)  
  "vendor/package": "^1.3.2", // >=1.3.2 <2.0.0  
  "vendor/package": "^0.3.2", // >=0.3.2 <0.4.0 // except if major version is 0
```

**COMPOSER + DRUPAL = ❤️**

# DÉPOTS COMPOSER POUR DRUPAL

Drupal.org possède maintenant ses propres dépôts composer :

<https://packages.drupal.org/7>

<https://packages.drupal.org/8>

Drupal core + modules tiers

# DRUPAL PROJECT : TEMPLATE PRÉ-CONFIGURÉ

<https://github.com/drupal-composer/drupal-project>

Installation de la dernière version stable de Drupal + Drush + Drupal console

Création et pré-configuration des dossiers (settings, files...)

Création d'un gitignore spécifique pour drupal



# MISE EN PLACE DE DRUPAL PROJECT

## Drupal 8

```
composer create-project drupal-composer/drupal-project:8.x-dev monsite.com --stability dev --no-interaction
```

## Drupal 7

```
composer create-project drupal-composer/drupal-project:7.x-dev monsite-d7.com --stability dev --no-interaction
```

# AJOUT DE DÉPENDANCES

```
composer require drupal/module
```

exemple : Devel

```
composer require drupal/devel
```

Utiliser le nom machine du module (comme avec drush) avec le vendor « drupal »

# UN MODULE SUR GITHUB ? 1/2

Ajouter l'adresse du dépôt dans la section « repositories »

```
composer config repositories.kgaut-tools vcs https://github.com/kgaut/kgaut_tools
```

Dans le composer.json :

```
"repositories": {  
  "0": {  
    "type": "composer",  
    "url": "https://packages.drupal.org/8"  
  },  
  "kgaut-tools": {  
    "type": "vcs",  
    "url": "https://github.com/kgaut/kgaut_tools"  
  }  
},
```

# UN MODULE SUR GITHUB ? 2/2

```
composer require kgaut/kgaut_tools
```

Note : le module devra posséder aussi son propre composer.json

```
{  
  "name": "kgaut/kgaut_tools",  
  "type": "drupal-module",  
  "description": "Tools and services for drupal 8",  
  "homepage": "https://github.com/kgaut/kgaut_tools",  
  "minimum-stability": "dev",  
  "require": { }  
}
```

# GESTION DES PATCHES

Dans la section extra du composer.json :

```
"patches": {  
  "drupal/core": {  
    "Entity display base hidden weight": "patches/entity-displaybase-hidden-weight.patch"  
  },  
  "drupal/simple_oauth" : {  
    "Allow emails to be used for auth": "https://www.drupal.org/files/issues/simple_oauth-allow-email-authentication-29"  }  
}
```

**POUR ALLER PLUS LOIN**

# LIENS ET RESSOURCES

Apprendre Git par l'exemple : <https://try.github.io/>

Problèmes et solutions de drupal + composer : <https://www.jeffgeerling.com/blog/2017/composer-and-drupal-are-still-st...>

Cette présentation : <https://slides.kgaut.net>

Quelques-unes de mes bafouilles sur composer : <https://kgaut.net/blog/tags/composer.html>

... et sur git : <https://kgaut.net/blog/tags/git.html>

Twitter : [@kgaut](#)

Mail : [contact@kgaut.net](mailto:contact@kgaut.net)