



```
$php->simplonBoulogne();
```

Composer





Sommaire

- Outil de gestion de dépendances
- Installation de Composer
- Intégration d'une dépendance
- Autoloading



Des librairies externes

Souvent des vos projets sous PHP, vous êtes amenés à devoir inclure plusieurs librairies (uploader de fichiers, test unitaires, formulaires, barre de recherche avancée etc...). La plupart de ces projets sont open-source.

Celles-ci contiennent donc des classes pré-codées qui peuvent donc être instanciées dans votre code



Composer

- Permet d'utiliser uniquement les librairies dont vous avez besoin
- Permet de mettre à jour facilement chaque librairies
- Composer trouve quelles librairies (et leurs versions) sont nécessaires pour installer une librairie spécifique
- Une organisation de vos librairies plus propre et conventionnée



Installation

Vous devez installer Composer pour chaque projet.

Commencez par télécharger le fichier `composer.phar` pour l'utiliser dans votre console.

Attention, la commande `curl` est nécessaire, je vous invite à faire une petite recherche Google si `curl` n'est pas installé par défaut ;)

```
curl -sS https://getcomposer.org/installer | php
```



Installation

Une astuce pour utiliser plus intuitivement composer est de déplacer le fichier `composer.phar` directement dans votre `PATH`.

Vous pourrez ensuite taper "`composer`" au lieu de "`php composer.phar`" pour utiliser composer ;)

```
mv composer.phar /usr/local/bin/composer
```



Initialisation

Permet de configurer le fichier `composer.json`, fichier lu par `composer` et qui structure les dépendances et leurs versions.

On peut bien évidemment modifier ce fichier manuellement pour ajouter vos dépendances.

```
composer init
```



Require

Permet de spécifier quelle dépendance vous souhaitez ajouter via la console. On doit par la suite indiquer un nom de package et sa version (en général la dernière) que vous souhaitez implémenter dans votre projet PHP.

Une fois terminé, votre fichier composer.json sera modifié.

```
composer require  
Search for a package:
```




composer.json

```
{
  "name": "maximesenecal/composer",
  "authors": [
    {
      "name": "Maxime Sénécal",
      "email": "msenecal@simplon.co"
    }
  ],
  "require": {
    "monolog/monolog": "^1.18"
  }
}
```



Autoloading

Voici l'intérêt principal de Composer :

Dans votre fichier PHP, placez au début de votre code dans le header.php ou index.php, l'autoloader de composer qui permet de charger toutes les classes en une ligne !

```
require 'vendor/autoload.php' ;
```



composer.lock

Le fichier `composer.lock` permet de verrouiller les versions nécessaires des librairies pour la bonne exécution du projet (par exemple au sein d'un projet d'équipe).

Si vous souhaitez mettre à jour le fichier `composer.lock` il faut faire la commande suivante :

```
composer update
```



composer.lock



Challenge



Rechercher et installer la librairie monolog (monolog/monolog sur Packagist).

Utiliser monolog pour créer facilement un fichier de log dans votre projet PHP.

Rechercher et installer la librairie swiftmailer (swiftmailer/swiftmailer sur Packagist).

Utiliser swiftmailer pour créer une formulaire de contact qui va envoyer un mail avec les champs remplis dans le formulaire PHP.

(Attention aux protocoles : SSL, TLS ... utilisés par votre mail server : Gmail, Yahoo)



Some resources

- <https://getcomposer.org/>
- <https://packagist.org/>
- <http://swiftmailer.org/docs/including-the-files.html>
- <http://www.sitepoint.com/sending-email-with-swift-mailer/>