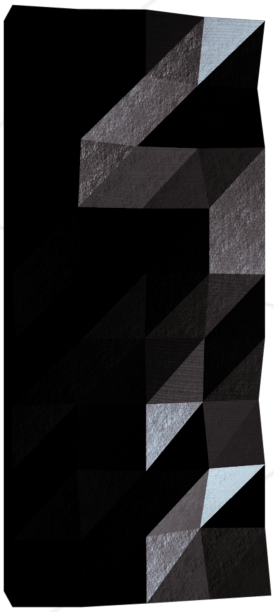

Formation PHP - Symfony

D03 - Composer

Timea LASZLO tlaszlo@pitechplus.com
Staff 42 bocal@staff.42.fr

Résumé: Lors de cette quatrième 42 journée de formation, vous allez découvrir le Composer et la façon dont vous pouvez l'utiliser dans vos applications.

Table des matières

Chapitre I

Préambule

Lorsque vous écrivez des applications en PHP, vous aurez probablement trouver que vous devez réinventer la roue à chaque fois que vous devez fournir une fonctionnalité commune comme la gestion des utilisateurs, la gestion de la base de données, etc. C'est là que le Composer peut vous sauver. Le git pullComposer est gestionnaire de dépendances qui va intégrer toutes les libraries nécessaires et dépendances pour les gérer au même endroit, dans votre projet. Ce type de gestion de dépendances dans un projet n'est pas un concept nouveau. En fait, beaucoup de fonctionnalités du Composer sont inspirées du NPM de Node.js ou encore du Bundler de Ruby. En comparaison avec d'autres gestionnaires de packages PHP, comme PEAR par exemple, le Composer va vous permettre d'installer les packages sur la base de chaque projet et pas au niveau global de vôtre installation PHP.

Chapitre II

Consignes

Sauf contradiction explicite, les consignes suivantes seront valables pour tous les jours de cette Piscine.

- Seul ce sujet sert de référence : ne vous fiez pas aux bruits de couloir.
- Le sujet peut changer jusqu'à une heure avant le rendu.
- Les exercices sont très précisément ordonnés du plus simple au plus complexe. En aucun cas nous ne porterons attention ni ne prendrons en compte un exercice complexe si un exercice plus simple n'est pas parfaitement réussi.
- Attention aux droits de vos fichiers et de vos répertoires.
- Vous devez suivre la procédure de rendu pour tous vos exercices. L'url de votre dépot `GIT` pour cette journée est disponible sur votre intranet.
- Vos exercices seront évalués par vos camarades de Piscine.
- En plus de vos camarades, vous pouvez être évalués par un programme appelé la Moulinette. La Moulinette est très stricte dans sa notation car elle est totalement automatisée. Il est donc impossible de discuter de sa note avec elle. Soyez d'une rigueur irréprochable pour éviter les mauvaises surprises.
- Les exercices shell doivent s'éxcuter avec `/bin/sh`.
- Vous ne devez laisser aucun autre fichier que ceux explicitement specifiés par les énoncés des exercices dans votre dépot de rendu.
- Vous avez une question ? Demandez à votre voisin de droite. Sinon, essayez avec votre voisin de gauche.
- Toutes les réponses à vos questions techniques se trouvent dans les `man` ou sur Internet.
- Pensez à discuter sur le forum Piscine de votre Intra et sur Slack !
- Lisez attentivement les exemples car ils peuvent vous permettre d'identifier un travail à réaliser qui n'est pas précisé dans le sujet à première vue.
- Réfléchissez. Par pitié, par Thor, par Odin !

Chapitre III

Règles spécifiques de la journée

Pas de règles spécifiques à cette journée.

Chapitre IV

Exercice 00

	Exercice : 00
Exercice 00 : Installer le Composer au niveau global	
Dossier de rendu : $ex00/$	
Fichiers à rendre : `Fichiers et répertoires de vôtre application`	
Fonctions Autorisées : `Toutes méthodes autorisées`	
Remarques : `n/a`	

Installez le **`Composer`** au niveau global, depuis le terminal.

Cet exercice est obligatoire. Afin de pouvoir résoudre tous les autres exercices vous devez absolument avoir résolu celui-ci.

Chapitre V

Exercice 01

	Exercice : 01
Exercice 01 : La spécification des versions	
Dossier de rendu : $ex01/$	
Fichiers à rendre : `Fichiers et répertoires de vôtre application`	
Fonctions Autorisées : `Toutes méthodes autorisées`	
Remarques : `n/a`	

Cet exercice comprend plusieurs tâches. Pour chacune d'entre elles vous devez créer un nouveau fichier composer.

Installez les versions suivantes du package **Monolog** en utilisant le **Composer**.

1. version 1.20.0
2. version supérieure à 1.19.0 et inférieure ou égale à 1.21.0
3. version entre 1.19 et 1.20
4. version supérieure ou égale à 1.19 et inférieure à 2.0
5. version supérieure à 1.19 et inférieure à 2.0

Astuce : `Afin de résoudre les tâches, vous devrez utiliser le commande` **`composer install`**.

Chapitre VI

Exercice 02

	Exercice : 02
Exercice 02 : Development requirement	
Dossier de rendu : $ex02/$	
Fichiers à rendre : `Fichiers et répertoires de vôtre application`	
Fonctions Autorisées : `Toutes méthodes autorisées`	
Remarques : `n/a`	

Installez la version 4.8.27 du package PHPUnit en tant que development requirement.

Chapitre VII

Exercice 03

	Exercice : 03
Exercice 03 : Composer install vs. composer update	
Dossier de rendu : $ex03/$	
Fichiers à rendre : `composer.json, composer.lock, fichiers et répertoires de vôtre application`	
Fonctions Autorisées : `Toutes méthodes autorisées`	
Remarques : `n/a`	

Cet exercice contient deux tâches principales. Pour chacune d'entre elles vous devez créer un sous-répertoire contenant les fichiers du `composer`.
Avec les fichiers **composer.json** et **composer.lock** dans vôtre répertoire courant, vous devrez exécuter les commandes **composer install** et **composer update**. Observez les fichiers créés/mis à jour et quelle est la différence entre `install` et `update`.
Marche à suivre :

1. Copier **`composer.json`** et **`composer.lock`** dans vôtre répertoire courant
2. Exécuter la commande **composer install**
3. Exécuter la commande **composer update**

Chapitre VIII

Exercice 04

	Exercice : 04
Exercice 04 : Ajouter un package via le Composer	
Dossier de rendu : *ex*04/	
Fichiers à rendre : `Fichiers et répertoires de vôtre application`	
Fonctions Autorisées : `Toutes méthodes autorisées`	
Remarques : `n/a`	

Créer un package personnalisé qui sera inclus dans les projets à l'aide du Composer. Le package doit implémenter une fonction qui retournera la météo actuelle de votre ville. Les conditions météo doivent être ajoutés dans un fichier **weather.txt** et doivent être exprimés en degrés Celsius.
Afin de créer être package vous devez utiliser un client REST qui fera appel à l'API. Vous devez trouver une API qui renvoie la météo courante de votre location actuelle. Il est possible que pour que vôtre demande météo par API soit acceptée vous devez vous enregistrer auprès du fournisseur de l'API. Ne pas ignorer cette étape.

Le package doit respecter les règles suivantes :

- le nom du package devrait avoir le format : **votreLogin/weather**
- il doit avoir requis la version 5.5.0 de **PHP**
- il doit avoir requis le package **RESTClient** pour PHP (https://github.com/tcdent/php-restclient)
- la stabilité doit être définie sur **dev**