Як встановити Composer

Composer можна встановити локально, директорії проекту, або глобально. Встановивши Composer глобально, його можна використовувати у будь-якому проекті.

Детальну інструкцію зі скачування інсталятора можна прочитати за цим посиланням. Після завантаження файлу установки можемо приступити до установки Composer.

Встановлення — Linux/Unix/macOS

Локально

Після правильного завантаження та встановлення інсталятора у директорії проекту з'явиться файл 'composer.phar'.

Щоб запустити Composer локально, достатньо виконати команду в терміналі, перебуваючи у директорії проекту.

```
php composer.phar
```

Глобально

Для того щоб зробити Composer глобальним і викликати його з будь-якої директорії, достатньо виконати команду в терміналі:

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

Тепер запустіть composer, щоб запустити менеджер пакетів замість php composer.phar.

Встановлення — Windows

Для того щоб встановити Composer глобально для Windows, потрібно виконати наступні кроки:

Створити новий composer.bat файл поруч з composer.phar:

Використовуючи cmd.exe:

```
C:\bin> echo @php "%~dp0composer.phar" %*>composer.bat
```

Використовуючи PowerShell:

```
PS C:\bin> Set-Content composer.bat '@php "%~dp0composer.phar" %*'
```

Додайте каталог до змінного середовища PATH, якщо це ще не зроблено. Для отримання інформації про зміну змінної PATH дивіться цю статтю.

Закрийте поточний термінал. У новому вікні терміналу введіть наступну команду, щоб перевірити роботу Composer:

```
C:\Users\username>composer -V
Composer version 2.0.12 2021-04-01 10:14:59
```

Синтаксис та опції Composer

Перше, що необхідно сказати, Composer — це консольна утиліта, вона не має графічного інтерфейсу, однак це не робить її гіршою. Її синтаксис досить простий, а подивитися всі опції та команди можна ввівши команду 'composer' у терміналі.

Список усіх опцій та команд можна розглянути нижче:

Найкорисніші:

- **-h** — вивести довідку по утиліті
- **-q** — скорочений варіант виведення
- **-V** — показати версію утиліти
- **-n** — не ставити інтерактивні питання
- **-v, -vv, -vvv** — налаштування подробиці виводу
- **-d** — використовувати вказану робочу директорію

Список команд, що часто використовуються:

- **archive** — архівує поточний проект як бібліотеку для відправки в мережу
- **check-platform-reqs** — перевіряє, чи дотримані системні вимоги
- **create-project** — створює проект на основі пакета в зазначену директорію
- **depends** — виводить залежності пакету

- **dump-autoload** — оновлює систему автозавантаження класів
- **exec** — дозволяє виконувати скрипти із встановлених пакетів
- **init** — створює порожній проект у поточній папці
- **list** — виводить список доступних команд
- **outdated** — виводить список пакетів, для яких є оновлення
- **prohibits** — виводить назви пакетів, які заважають встановити вказаний пакет
- **search** — пошук пакетів у репозиторіях
- **self-update** — оновлення Composer до останньої версії, працює тільки при локальній установці
- **show** — інформація про пакет
- **update** — оновлює всі пакети до найактуальнішої версії

Встановлення проекту на основі пакету

Швидше за все, для майбутніх проектів, ви використовуватимете фреймворки, щоб мати базовий функціонал з коробки. У таких випадках вам знадобиться розгортати проект на основі існуючого пакета, Composer завантажить пакет з репозиторію і просто розпакує його в потрібну вам директорію.

Для цього використовуйте команду **create-project**, наприклад:

```
composer create-project laravel/laravel app_dir
```

де:

- **composer** — звернення до утиліти
- **create-project** — команда
- **laravel/laravel** — пакет фреймворку laravel
- **app_dir** — директорія, в яку Composer розпакує вказаний пакет

Встановлення пакетів

Щоб інсталювати пакет за допомогою Composer, необхідно використовувати команду **require**. Утиліта встановить вказаний пакет і запише його у файл composer.json.

Наприклад, встановимо пакет для зручного виведення змінних:

```
composer  require  larapack/dd
```

Також можна вказати опцію --dev перед написанням назви пакета, щоб пакет був обов'язковим, але використовувався лише у розробці.

Усі доступні пакети можна переглянути в сервісі Packagist.

Оновлення пакетів

Пакети необхідно оновлювати, щоб уникнути уразливостей у проектах та усунення старих проблем з боку підключених бібліотек.

Для оновлення всіх пакетів достатньо ввести команду **update**:

```
composer  update
```

або

```
composer  update –dev
```

Видалення пакетів

Для видалення пакета необхідно ввести команду **remove** та вказати потрібний пакет, також можна додати опцію –dev, якщо це пакет для розробників.

```
composer  remove  larapack/dd
```

Можна також видалити непотрібні пакети у файлі composer.json та запустити команду на оновлення.

Скидання автозавантаження

Composer з коробки може бути ролі автозавантажувача класів. Він використовує стандарт PSR-4. Іноді трапляється, що класи закешувалися, і новий клас не помітний у проекті.

У таких випадках необхідно виконати команду скидання:

```
composer  dump-autoload
```

або

```
composer  du
```

Висновки

Composer це досить потужна утиліта для того, щоб зробити вашу роботу з проектом комфортною. З її допомогою ви легко зможете додавати, оновлювати та видаляти необхідні вам пакети. Також приємним бонусом є те, що утиліта надає автозавантажувач класів із коробки.