



Composer++

Руслан, Ханов

Разработчик, Университет ИТМО

Активист сообщества Symcode

План

- | что будет в докладе
- | чего не будет в докладе

План

- › введение
- › как работает
- › проблемы
- › решения
- › composer 2.0
- › ВЫВОДЫ

Введение в composer

Альтернативы

- | Phing
- | Makefile
- | Gradle

Альтернативы, есть но ...

- | нативность
- | богатый набор фич
- | качество
- | популярность

Введение

| Composer - это ...

- › менеджер пакетов для PHP
- › менеджер зависимостей для PHP

| Composer это ...

- › управление пакетами для PHP
- › управление зависимостями для PHP

Пакеты или зависимости

- › Пакет - можно сделать больше.
- › Зависимость - нельзя сделать нужное.

Как работает Composer

Схема работы: Компоненты

Компоненты

- › консольное приложение
- › автолоадер
- › логгер
- › окружение запуска команд
- › плагины

Схема работы без плагинов



Плагин

`composer-plugin` пакет

шина событий

набор методов

1. `PluginInterface::activate`
2. `PluginInterface::deactivate`
3. `PluginInterface::uninstall`
4. `EventSubscriberInterface::getSubscribedEvents`

Схема работы с плагинами



Популярные плагины

- › hirak/prestissimo
- › brainmaestro/composer-git-hooks
- › mouf/nodejs-installer
- › neronmoon/scriptsdev
- › symfony/flex*

использую за исключением *

Экзотические плагины

- › clue/graph-composer
- › wikimedia/composer-merge-plugin
- › slince/composer-registry-manager
- › Composer Patches

изучал, но предпочел не использовать

Примеры использования плагинов

| composer require brainmaestro/composer-git-hooks

```
"extra": {  
  "hooks": {  
    "pre-commit": "my-mighty-one-liner.sh"  
  }  
}
```

| cat .git/hooks/pre-commit

```
#!/bin/sh
```

```
my-mighty-one-liner.sh
```


Проблемы

Проблемы

1. много скриптов
2. вывод результатов выполнения команды
3. переменные окружения
4. фоновые процессы
5. редактирование всех настроек из консоли (Composer 2.0*)
6. оффлайн-режим (Composer 2.0)
7. повторный запуск после успешного шага

* - только секция extra

Фоновый процесс, пример

```
"scripts": {  
  "report": "firefox report.html"  
}
```

Фоновый процесс, костыль

```
"scripts": {  
  "report": "(nohup ... 1>>/dev/null 2>&1 &); true"  
}
```

Решения

Много скриптов, решение

```
class ScriptsCommandProvider implements CommandProvider
{
    public function getCommands()
    {
        $commands = [];
        /** fullfill $commands */
        return $commands;
    }
}
```

Вывод результатов выполнения команды, решение

- | Интеграция с `monolog/monolog` или аналогами
или
- | Пропатчить компонент `symfony/console`

```
class CustomVerbosityCommand extends BaseCommand {  
    public function run(** */) {  
        // $output->write(** *);  
        $this->writeWithCustomVerbosity(** *);  
    }  
  
    protected function writeWithCustomVerbosity(** *) {  
        $customVerbosity = $this->getVerbosity($defaultVerbosity);  
        $output->write(  
            $message,  
            $newline,  
            $customVerbosity  
        );  
    }  
}
```


Переменные окружения, решение

`init` событие

`extra` для реестра переменных

`getenv` И `putenv`

Фоновый процесс, решение

- | & или `nohup` или `dosown`
- | уход в фоновый процесс на уровне запускаемого скрипта
- | замена `symfony/process` на `noname/background-process`

Редактирование composer.json из консоли, решение

| своя схема для composer.json

Повторный запуск после успешного шага, решение

- | решения не знаю (
- | но знаю где подсмотреть (Gradle)

Composer 2.0

Composer 2.0

1. `hirak/prestissimo` больше не нужен
2. `--dry-run` для всех команд управления пакетами
3. `dump-autoload` -> `check-platform-reqs`
4. `--no-suggest` удалили?!
5. `extra` полностью из консоли
6. работа с репозиториями
7. оффлайн-режим

Выводы

Выводы

1. используйте скрипты
2. используйте плагины
3. форк composer-а? Может быть не надо?!

Полезная ссылка

› <https://github.com/jakoch/awesome-composer>



Вопросы?

 [hanovruslan](#)

 [hanovruslan](#)

 hanov.ruslan@gmail.com

Руслан, Ханов

Разработчик, Университет ИТМО

Активист сообщества Symcode