Technologia i rozwiązania

Laravel 4
Podstawy tworzenia aplikacji w PHP

Przewodnik dla początkujących!

Raphaël Saunier

Helion

[PACKT] open source
PUBLISHING
community experience distilled

Tytuł oryginału: Getting Started with Laravel 4

Tłumaczenie: Rafał Jońca

ISBN: 978-83-283-0298-3

Wydawnictwo HELION
ul. Kościuszki 1c, 44-100 GLIWICE
tel. 32 231 22 19, 32 230 98 63
e-mail: *helion@helion.pl*
WWW: *http://helion.pl* (księgarnia internetowa, katalog książek)

Pliki z przykładami omawianymi w książce można znaleźć pod adresem:
ftp://ftp.helion.pl/przyklady/larave.zip

Drogi Czytelniku!
Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres
http://helion.pl/user/opinie/larave
Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

Printed in Poland.

Spis treści

ROZDZIAŁ 2

Narzędzie Composer

W poprzednim rozdziale dowiedzieliśmy się, że Laravel powstał na bazie kilku pakietów niezależnych twórców. Zamiast dołączyć zewnętrzne zależności do własnego kodu źródłowego, Laravel korzysta z narzędzia **Composer**, zarządcy zależności, który pobiera i aktualizuje wszystkie pakiety. Z punktu widzenia narzędzia Laravel jest po prostu jeszcze jedną zależnością, którą należy zainstalować.

W tym rozdziale omówimy następujące tematy:

- problemy rozwiązywane przez zarządcę zależności,
- instrukcje instalacji zarządcy w systemach Windows oraz Unix (Mac OS X i Linux),
- tworzenie nowego projektu Laravel za pomocą Composer,
- znajdowanie i instalacja pakietów dodatkowych,
- ogólne wskazówki związane z korzystaniem z Composer.

Composer został zainspirowany przez innych popularnych zarządców zależności z innych języków programowania, np. Bundler ze społeczności Ruby lub npm wykorzystywany przez **node.js** oraz przeglądarkowe projekty JavaScript, i stara się zapewnić podobną funkcjonalność językowi PHP. Domyślnie pakiety nie są instalowane globalnie — podstawowy sposób działania to instalacja pakietów na potrzeby konkretnego projektu. Jeśli pewne zależności trzeba zainstalować w całym systemie PHP, skorzystaj z **PEAR** — zarządcy pakietów dostarczanego domyślnie wraz z PHP.

Oto kilka podstawowych zalet korzystania z zarządcy zależności we własnym projekcie:

- obsługa staje się znacznie szybsza, bo nie trzeba samodzielnie szukać, pobierać i rozpakowywać poszczególnych pakietów;
- unika się konfliktów wersji w trakcie aktualizacji poszczególnych zależności;
- wczytywanie poszczególnych klas odbywa się automatycznie bez udziału programisty;
- odkrywanie i wybór odpowiednich pakietów okazuje się znacznie łatwiejsze dzięki centralnemu repozytorium.

Korzystanie z wiersza poleceń

Jeśli dopiero zaczynasz swoją przygodę z programowaniem, prawdopodobnie nigdy wcześniej nie miałeś do czynienia z **interfejsem wiersza poleceń** (**CLI**). Korzystanie z narzędzia Composer, a w dalszej części książki również z narzędzia **Artisan** specyficznego dla Laravel, wymaga znajomości przynajmniej podstaw obsługi wiersza poleceń.

Oto opis sposobu uruchomienia wiersza poleceń w poszczególnych systemach.

1. W systemie Windows poszukaj programu *Wiersz polecenia*. Jeśli go nie znajdziesz, użyj polecenia *Start/Uruchom* (w systemie Windows 8.1 kliknij ikonę Windows prawym klawiszem myszy) i wpisz w polu tekst `cmd.exe`.
2. W systemie Mac OS X wiersz poleceń nosi nazwę *Terminal*. Znajdziesz go w folderze */Applications/Utilities*.
3. W systemie Linux nazwa zależy od dystrybucji. Najczęściej nosi ona nazwę *Terminal* lub *Konsole*. Większość osób korzystających z tego systemu używa wiersza poleceń stosunkowo często.

Obsługa Laravel opisywana w książce nie wymaga żadnej zaawansowanej wiedzy na temat wiersza poleceń. Wystarczy znajomość sposobu przedostania się do odpowiedniego folderu w systemie plików przed wykonaniem wskazanych poleceń. W tym celu wpisz polecenie `cd`, a następnie podaj ścieżkę docelową.

> W większości systemów można również wpisać polecenie cd, dodać znak spacji, a następnie przeciągnąć odpowiedni folder do okna terminalu.

W systemie Windows należy wykonać polecenie podobne do wskazanego poniżej.

```
> cd C:\ścieżka\do\folderu\z\kodem
```

Jeżeli jakiś przykład będzie dotyczył nie tylko systemu Windows, to w dalszej części książki do oznaczenia polecenia wykorzystamy znak `$`, a poszczególne foldery będą oddzielane znakiem ukośnika. Pamiętaj o samodzielnym dostosowaniu polecenia do systemu Windows.

Jak działa Composer?

Composer (*http://getcomposer.org/*) to plik wykonywalny PHP dodawany do zmiennej środowiskowej `PATH` (zawiera listę folderów, w których system szuka poleceń do wykonania). Po odpowiedniej instalacji z dowolnego folderu można wywołać polecenie `composer` uruchamiające zarządcę pakietów. Projekt wraz z zależnościami jest zdefiniowany w pliku JSON o nazwie *composer.json*. Composer odczytuje zawartość pliku i łączy się z internetowym repozytorium **Packagist** (*https://packagist.org/*), aby w sposób rekurencyjny pobrać oraz rozwiązać wszystkie zależności.

Następnie zależności są pobierane do lokalnego folderu (w Laravel jest to folder *vendor/*), a aktualny stan zależności zostaje zapisany w pliku *composer.lock*. Composer generuje również plik w folderze *vendor/*, który zawiera obsługę automatycznego wczytywania klas jako skrypt PHP (we właściwym kodzie aplikacji skrypt uruchamia polecenie `require 'vendor/autoload.php'`).

Instalacja

W systemach typu Unix i w systemie Windows instalacja Composer nie jest trudna, gdyż udostępniono instalator lub skrypty instalacyjne.

Unix (Mac OS X, Linux)

Przede wszystkim musimy się upewnić, że plik wykonywalny `php` jest dostępny z poziomu wiersza poleceń. W tym celu otwórz okno terminalu i wpisz następujące polecenie.

```
$ php -v
```

Powinna pojawić się informacja na temat wersji PHP aktualnie zainstalowanej w systemie. Jeśli pojawi się błąd informujący o nieznanym poleceniu lub też wersja zainstalowanego PHP okaże się zbyt niska (minimum to wersja 5.3.7), skorzystaj ze wskazówek dotyczących instalacji PHP w danym systemie dostępnych na stronie *http://php.net*. Jeśli w systemie Mac OS do instalacji użyto **MAMP** lub **Homebrew**, upewnij się, że zmienna środowiskowa `PATH` zawiera w pierwszej kolejności zainstalowaną wersję zamiast domyślnej wersji z systemu operacyjnego.

Gdy mamy pewność, że odpowiednia wersja PHP jest dostępna z poziomu wiersza poleceń, należy zainstalować Composer za pomocą poniższego polecenia:

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

Aby instalacja była dostępna z poziomu dowolnego folderu, przenieś ją do katalogu znajdującego się w zmiennej `PATH`. Jeśli pojawi się błąd dostępu, wybierz inny folder, do którego masz prawo zapisu. Ewentualnie skorzystaj z polecenia `sudo`.

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

Aby mieć pewność, że wszystko zostało zainstalowane prawidłowo, ponownie otwórz okno terminalu i wpisz następujące polecenie:

```
$ composer
```

Spowoduje to wyświetlenie listy wszystkich dostępnych poleceń narzędzia.

Windows

Aby zainstalować Composer w systemie Windows, wystarczy pobrać plik *Composer-Setup.exe* dostępny pod adresem *https://getcomposer.org/download/*. Oczywiście zadziała on tylko wtedy, gdy jest zainstalowana odpowiednia wersja PHP.

Instalator poprosi o wskazanie lokalizacji pliku wykonywalnego PHP. Oto kilka typowych lokalizacji:

- domyślna — *C:\PHP5\php.exe* lub *C:\PHP\php.exe*,
- przy użyciu XAMPP — *C:\xampp\php\php.exe*,
- przy użyciu WAMP — *C:\wamp\bin\php\php5.x.x\php.exe*.

Jeśli plik wykonywalny nie znajduje się w przedstawionych lokalizacjach, ale masz pewność, że został zainstalowany w systemie, użyj narzędzia wyszukiwania, aby znaleźć plik *php.exe*.

Instalator zajmie się pobraniem i zainstalowaniem narzędzia Composer, doda także polecenia `php` i `composer` do zmiennej środowiskowej `PATH`.

Aby się upewnić, że instalacja przebiegła pomyślnie, otwórz nowy wiersz poleceń i wpisz dwa polecenia przedstawione poniżej.

```
> php -v
> composer
```

Oba polecenia powinny wykonać się bez błędów, a następnie wyświetlić informację o wersji oraz dostępnych poleceniach narzędzia.

Tworzenie nowej aplikacji Laravel

Po zainstalowaniu narzędzia Composer utworzenie nowego projektu Laravel jest dziecinnie proste. Po przejściu do folderu, w którym chcesz utworzyć projekt, wykonaj następujące polecenie:

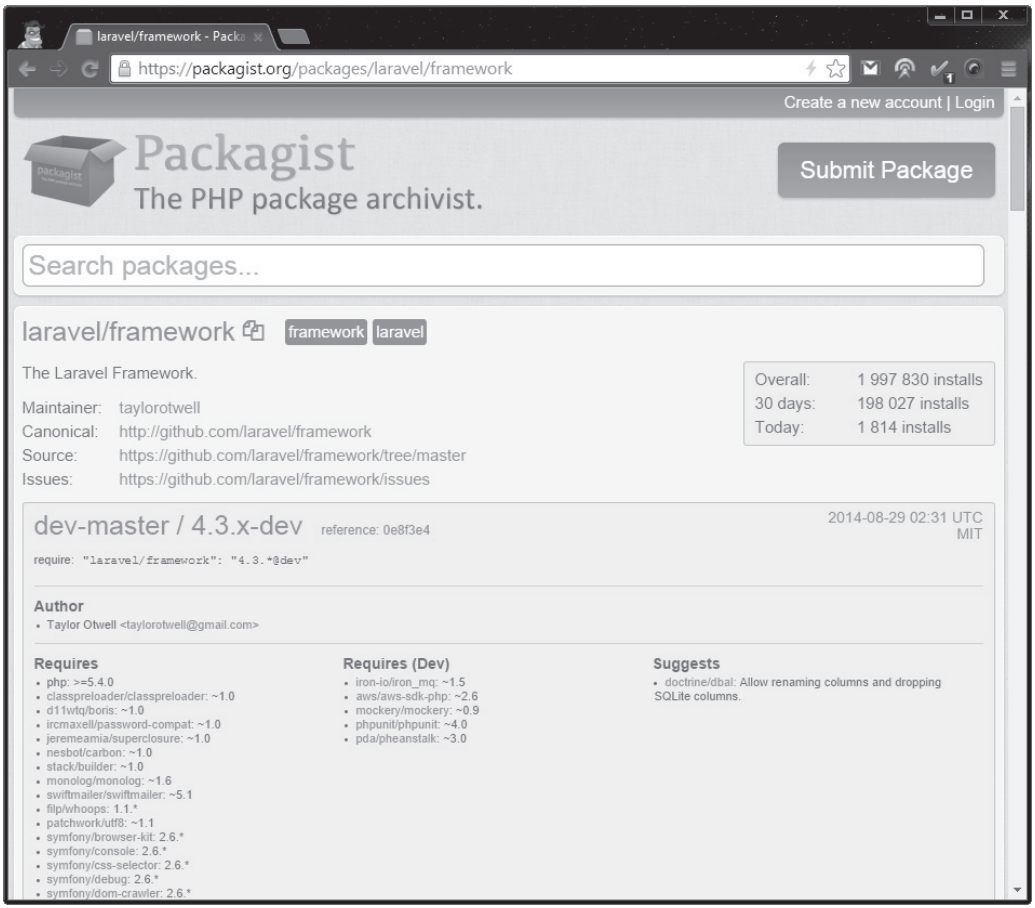
```
$ composer create-project laravel/laravel --prefer-dist
```

Spowoduje to pobranie najnowszej wersji Laravel wraz z zależnościami. Ponieważ szybkość łącza internetowego i moc procesora wpływają na pobranie i instalację, ta może zająć nawet kilka minut. Po zakończeniu całego procesu w folderze pojawi się cała struktura katalogów opisana w poprzednim rozdziale oraz gotowa do uruchomienia projektu.

Jeśli korzystasz z systemu kontroli wersji Git, warto w tym momencie uruchomić polecenie `git init`. Główny folder zawiera odpowiednio przygotowany plik *.gitignore*, a puste foldery zawierają pliki *.gitkeep*.

Odnajdywanie i instalacja nowych pakietów

Strona wyszukiwania dostępna pod adresem *http://packagist.org* ułatwia odnalezienie i dodanie pakietów oferujących przydatne funkcje, np. obróbkę obrazów lub generowanie plików PDF. Wskaźnikami, czy pakiet wart jest uwagi, są nie tylko liczba pobrań i liczba gwiazdek w serwisie GitHub, ale również jakość dokumentacji, pokrycie testami lub ogólna aktywność projektu. Przed dodaniem pakietu można na stronie Packagist sprawdzić jego różne wersje i zależności.



W trakcie prac programistycznych można z powodzeniem stosować wersję dev-master, ale przed udostępnieniem wersji produkcyjnej warto wskazać konkretny numer wersji, aby uniknąć potencjalnych problemów ze zgodnością.

Aby zainstalować pakiet, w edytorze tekstu otwórz plik *composer.json* i w obiekcie `require` wstaw nazwę pakietu oraz wymaganą wersję.

```
"require": {
   "laravel/laravel": "v4.1.*",
   "intervention/image": "dev-master"
}
```

Ponieważ jest to plik JSON, uważaj, aby przypadkiem nie pozostawić na końcu żadnych zbędnych znaków przecinka. Użyj polecenia `composer validate`, aby sprawdzić, czy plik *composer.json* nie zawiera żadnych błędów.

Następnie wykonaj polecenie `composer install`, a Composer pobierze pakiet i wszystkie jego zależności. Aby zaktualizować plik *composer.lock* i zapisać dokładne numery wersji poszczególnych zależności, wykonaj polecenie `composer update`. W momencie wdrożenia lub rozpowszechniania aplikacji plik z wersjami umożliwi innym osobom pobranie za pomocą polecenia `composer install` dokładnie tych samych wersji pakietów. Polecenie `update` zawsze poszukuje najnowszych wersji każdego pakietu, co w przypadku systemu produkcyjnego może doprowadzić do błędów związanych ze zgodnością między pakietami.

Kilka dodatkowych rad

Przed rozpoczęciem tworzenia pierwszej aplikacji Laravel warto zaznajomić się z kilkoma wskazówkami pomagającymi w codziennym korzystaniu z narzędzia Composer.

- Umieść plik *composer.lock* w systemie kontroli wersji, aby uniknąć pobierania niezgodnych ze sobą wersji pakietów i upewnić się, że wszyscy programiści korzystają z pakietów w dokładnie tych samych wersjach.
- Nie trzeba umieszczać zawartości folderu *vendor/* w systemie kontroli wersji. Domyślnie jest on wyłączony z systemu kontroli wersji Git za pomocą odpowiedniego wpisu w pliku *.gitignore*. Dodanie go do systemu kontroli wersji zwiększyłoby jedynie rozmiar repozytorium bez żadnych istotnych korzyści. Dopóki repozytorium zawiera pliki *composer.json* i *composer.lock*, każdy będzie mógł odtworzyć tę samą zawartość folderu z pakietami.
- Nie należy edytować plików znajdujących się w folderze *vendor/*, ponieważ mogą zostać automatycznie nadpisane przy następnym użyciu polecenia `composer install`.
- Composer w trakcie instalacji pakietu udostępnia dwie opcje — `--prefer-source` i `--prefer-dist`. Różnica między nimi polega na tym, że w przypadku drugiej z nich Composer będzie preferował pobieranie wydań stabilnych i w miarę możliwości nie będzie pobierał całej historii z repozytorium Git.
- Łączny rozmiar folderu *vendor/* będzie oscylował w granicach 25 MB w przypadku opcji `--prefer-dist` i mniej więcej trzykrotnie więcej w przypadku opcji `--prefer-source`, ponieważ zostanie pobrana pełna historia z repozytorium Git. Po umieszczeniu aplikacji Laravel na serwerze należy na nim uruchomić polecenie `composer install`. Jeśli jedyną dostępną opcją jest przesyłanie wszystkich plików za pomocą protokołu

FTP, skorzystaj z pakietów takich jak `barryvdh/laravel-vendor-cleanup`, które usuną wszystkie zbędne pliki przed umieszczeniem aplikacji na serwerze.

- Polecenie `diagnose` narzędzia Composer oraz opcje dodatkowych informacji (`-v`, `-vv` i `-vvv`) pomogą zidentyfikować typowe problemy i poprosić o pomoc na forum, `IRC` lub w witrynie **Stack Overflow**.

Podsumowanie

W tym rozdziale wyjaśniliśmy, jakie problemy rozwiązuje użycie zarządcy zależności. Zainstalowaliśmy narzędzie Composer i utworzyliśmy pierwszy projekt Laravel. Dowiedzieliśmy się również, w jaki sposób znaleźć i zainstalować pakiet oraz jak uniknąć typowych błędów związanych z korzystaniem z narzędzia Composer.

W następnym rozdziale rozpocznie się prawdziwa zabawa! Po uzyskaniu w pełni działającej wersji narzędzia Composer przejdziemy przez wszystkie kroki niezbędne do utworzenia w pełni działającej aplikacji Laravel.

Skorowidz

G

H

I

J

K

L

Ł

M

N

O

P

Z

Laravel 4
Podstawy tworzenia aplikacji w PHP

Laravel to szkielet aplikacji dla języka PHP, zdobywający w ostatnim czasie ogromną popularność. Powody są dwa: obszerna i przejrzysta dokumentacja oraz podstawowe założenie autorów, że tworzenie aplikacji z wykorzystaniem Laravela ma być po prostu przyjemne. I tak jest! Łatwe zarządzanie bazą danych, przyjazny system szablonów, intuicyjna konfiguracja to tylko niektóre z zalet tego szkieletu.

Jeżeli chcesz poznać w pełni jego potencjał i wykorzystać go w trakcie tworzenia kolejnej strony WWW lub aplikacji internetowej, to trafiłeś na doskonałą książkę. Sięgnij po nią i już wkrótce bez problemu przygotujesz swoje środowisko pracy, zainstalujesz najnowszą wersję za pomocą narzędzia Composer, a następnie rozpoczniesz prace nad swoim projektem. Przekonaj się, jak tworzyć ścieżki, konfigurować bazę danych oraz zabezpieczać dostęp do wybranych stron. Dzięki lekturze kolejnych rozdziałów nauczysz się tworzyć testy automatyczne oraz korzystać z narzędzia Artisan, które ułatwi Ci codzienne zadania! Ponadto dowiesz się, w jaki sposób tworzyć serwisy typu REST oraz generować formularze. Książka ta jest idealną pozycją dla wszystkich programistów PHP szukających frameworka, który spełni ich oczekiwania!

Dzięki tej książce:

- poznasz narzędzie Composer
- pobierzesz najnowszą wersję szkieletu Laravel
- przygotujesz ścieżki dostępu do stron
- przetestujesz stworzony kod
- błyskawicznie zbudujesz pierwszą aplikację WWW

Tworzenie aplikacji w PHP jeszcze nigdy nie było takie proste!

[PACKT] open source* PUBLISHING community experience distilled

Helion

27583 numer katalogowy

księgarnia internetowa

http://helion.pl

zamówienia telefoniczne

0 801 339900

0 601 339900

Sprawdź najnowsze promocje:
http://helion.pl/promocje
Książki najchętniej czytane:
http://helion.pl/bestsellery
Zamów informacje o nowościach:
http://helion.pl/nowosci

Helion SA
ul. Kościuszki 1c, 44-100 Gliwice
tel.: 32 230 98 63
e-mail: helion@helion.pl
http://helion.pl

ISBN 978-83-283-0298-3

cena: 34,00 zł