
LARAVEL

- PHP - найбільш популярна в світі серверна скриптова мова. Вона пройшла великий шлях розвитку від невеликих, вбудованих в код статичних HTML сторінок, сніпетів, до сучасної мови, на якому розробляється більшість сучасних динамічних сайтів. Складність і функціональність сучасних сайтів тільки зростає, і ні у кого немає бажання писати весь необхідний код з нуля. Програмістам необхідно розробляти складні сайти і веб-додатки, а це зазвичай займає дуже багато часу. Щоб полегшити процес розробки програмістам, почали створюватися фреймворки.

- Laravel - це фреймворк для web-додатків з виразним і елегантним синтаксисом. Він дозволить спростити вирішення основних наболілих завдань, таких як аутентифікація, маршрутизація, сесії і кешування. Laravel - це спроба об'єднати все найкраще, що є в інших PHP фреймворках. Основні переваги Laravel:

- Велика екосистема з миттєвим розгортанням своєї платформи. Офіційний сайт надає безліч мануалів і інформації для ознайомлення;

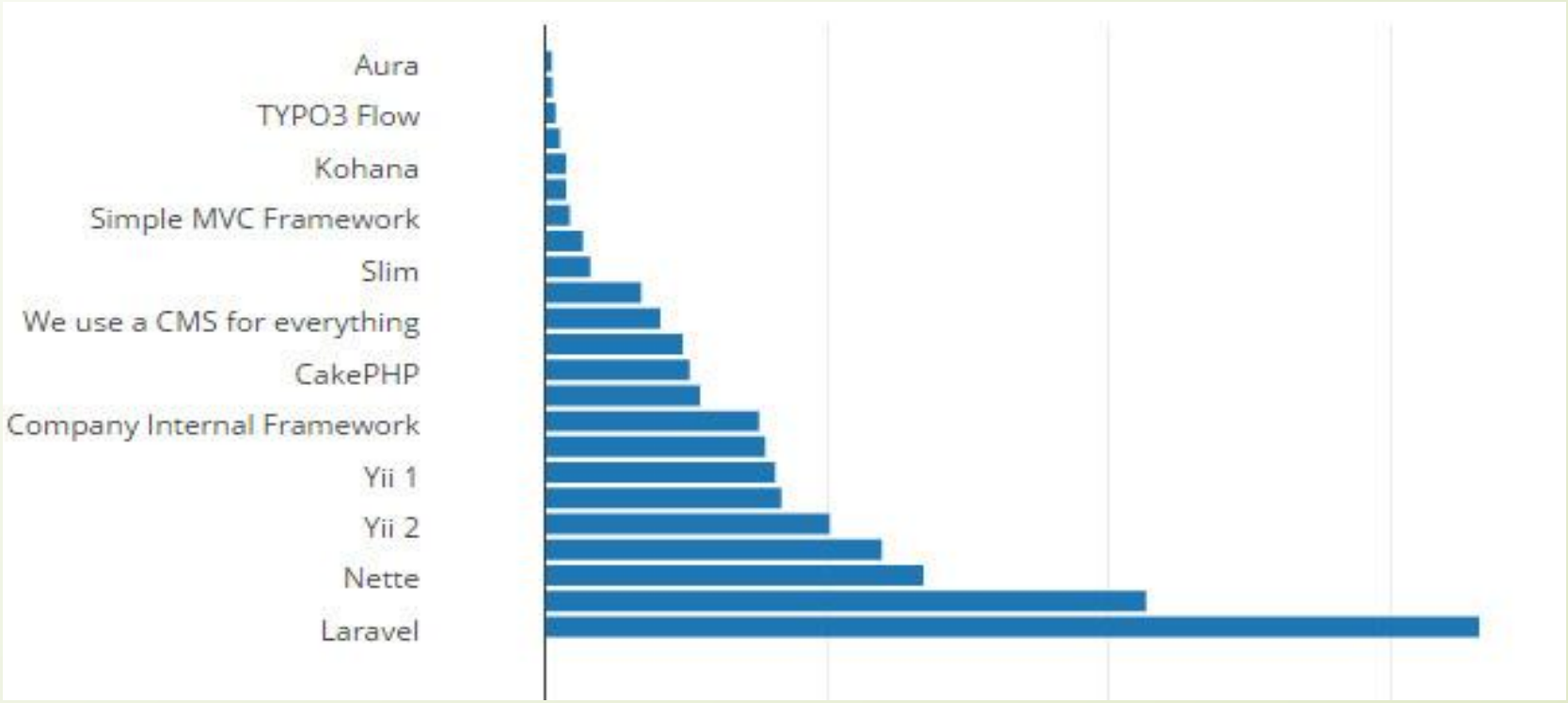
- Документація Laravel близька до досконалості;

- У Laravel є свій движок для шаблонів Blade, «гарний» синтаксис мови, який сприяє вирішенню всіх необхідних завдань, таких як аутентифікація, сесії, кешування і маршрутизація RESTful.

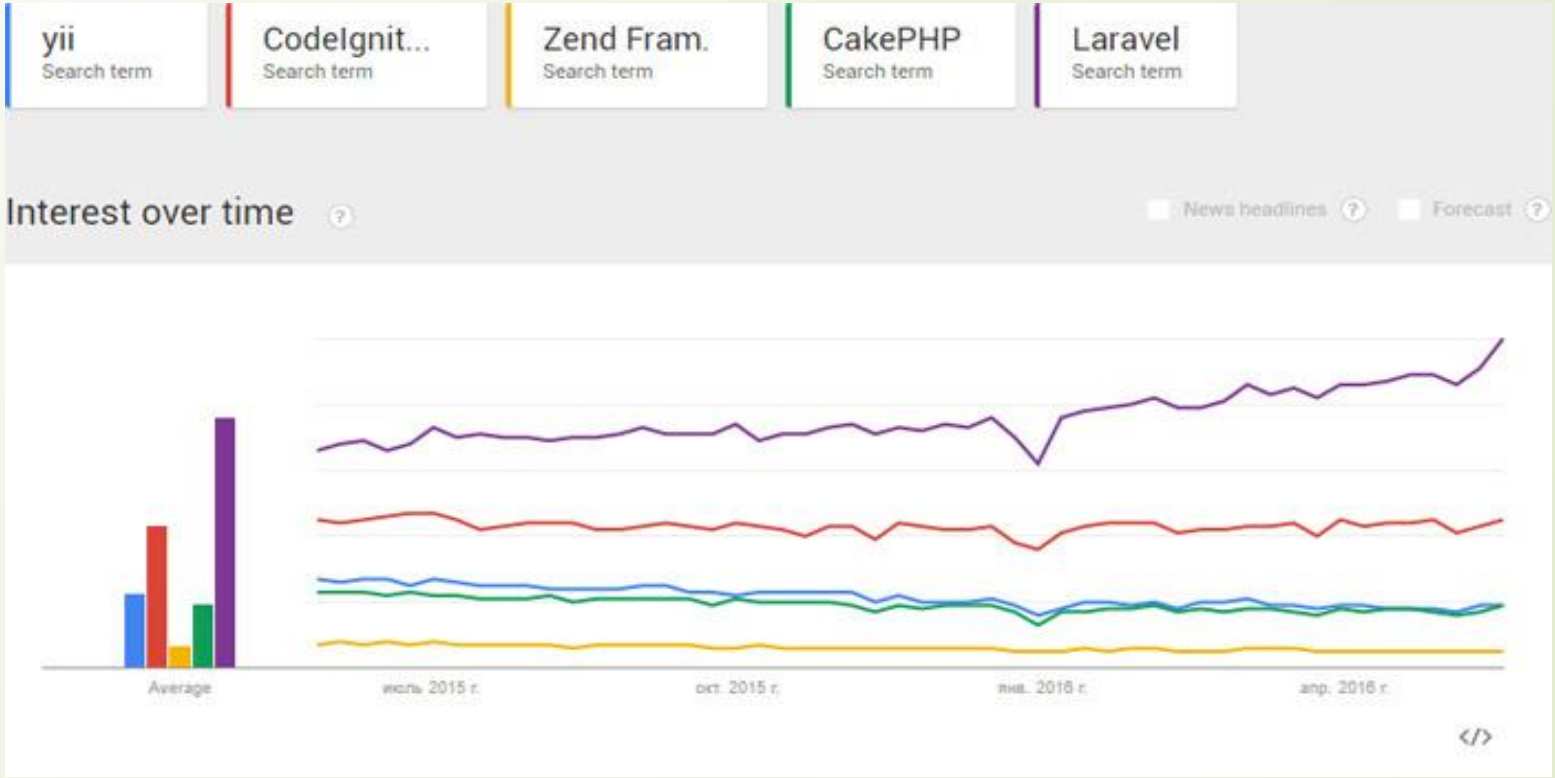
- Незважаючи на свою молодість (перший реліз вийшов в 2011 році), це вже зовсім зрілий продукт, і, згідно з опитуванням, проведеним порталом SitePoint, він займає перше місце за популярністю серед розробників на PHP.

Популярність фреймворку Laravel

? Для того щоб оцінити популярність фреймворку, пропоную звернутися до декількох джерел статистики використання. Статистика популярності від Sitepoin за 2015 рік.

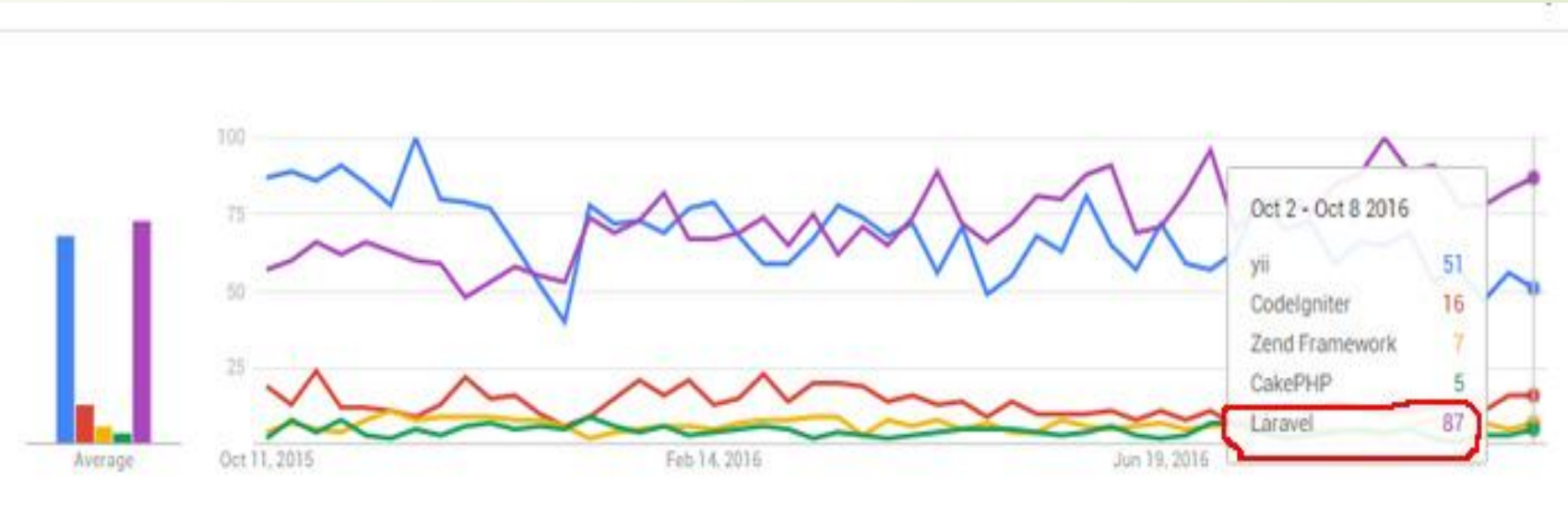


? Популярність за статистикою запитів, яку надає Google Trends.



- Як видно з графіків і статистики від SitePoint, а також зі статистикою Google Trends, фреймворк Laravel займає впевнене перше місце і постійно набирає популярність, не зменшуючи обертів. Що, загалом, дуже добре, враховуючи, що це загальносвітова статистика.

- Якщо ми візьмемо статистику запитів в Google Trends по Росії, то картинка дещо зміниться:



- Але з графіка очевидний факт, що протягом останніх 4 місяців фреймворк Laravel утримує позиції лідера і продовжує набирати популярність.

- Це означає, що більшість нових проектів в країні тепер створюються з використанням активно-розвиваючогося Laravel. З огляду на тренди, якщо ви давно хотіли почати вчити Laravel, то зараз той самий момент, коли це пора зробити.

Можливості Laravel

- Пакети - дозволяють створювати і підключати модулі в форматі Composer до додатка на Laravel. Багато додаткових можливостей вже доступні у вигляді таких модулів.

- Eloquent ORM - реалізація шаблону проектування ActiveRecord на PHP. Дозволяє строго визначити відносини між об'єктами бази даних. Стандартний для Laravel будівник запитів Fluent підтримується ядром Eloquent.

- Логіка додатка - частина розробляючогося додатку, оголошена або за допомогою контролерів, або маршрутів. Зворотня маршрутизація пов'язує між собою згенеровані додатком посилання і маршрути, дозволяючи змінювати останні з автоматичним оновленням пов'язаних посилань. При створенні посилань за допомогою іменованих маршрутів Laravel автоматично генерує кінцеві URL.

- REST-контролери - додатковий шар для поділу логіки обробки GET- і POST-запитів HTTP.

- Автозавантаження класів - механізм автоматичного завантаження класів PHP без необхідності підключати файли їх визначень в include. Завантаження на вимогу запобігає завантаженню непотрібних компонентів; завантажуються тільки ті з них, які дійсно використовуються.

- Укладачі уявлень - блоки коду, які виконуються при генерації уявлення (шаблону).

- Інверсія управління - дозволяє отримувати екземпляри об'єктів за принципом зворотного управління. Також може використовуватися для створення і отримання об'єктів-одинаків.

- Міграції - система управління версіями для баз даних. Дозволяє пов'язувати зміни в коді програми зі змінами, які потрібно внести в структуру БД, що спрощує розгортання і оновлення програми.

- Модульне тестування (юніт-тести) - грає дуже велику роль в Laravel, який сам по собі містить велику кількість тестів для запобігання помилок.

- Сторінковий вивід - спрощує генерацію сторінок, замінюючи різні способи вирішення цієї задачі єдиним механізмом, вбудованим в Laravel.

Особливості фреймворку

- Laravel - це особливий фреймворк з сильним брендингом, тому майже все особливе в Laravel має унікальну назву.

- Dotenv

Laravel використовує добре відомий файл .env для зберігання секретної інформації, такої як пароль від БД, логін для email та інші речі. Це файл, в якому ви визначаєте пари ключ-значення для будь-якої вашої секретної інформації.

- PSR-4

Laravel - перший фреймворк, що підтримує PSR-4. Прямо з коробки Composer автоматично завантажить всі класи з каталогу app, використовуючи стандарт автозавантаження PSR-.

Це означає, що у вас може бути один простір імен для вашого застосування, і ви можете структурувати його, як захочете, і як вважаєте за логічним. Laravel не вимагає від вас розміщення певних файлів в певних папках.

- IoC-контейнер - розширення ядра

Контейнер зворотного управління в Laravel - потужний інструмент для управління залежностями класів. Впровадження залежностей це спосіб зняття жорстко закодованих залежностей класу. Замість цього залежності впроваджуються під час виконання, забезпечуючи більшу гнучкість, так як реалізація залежностей може бути легко змінена.

Ви можете використовувати IoC-контейнер в вашому додатку, щоб зробити все що впроваджуються і легко тестується за допомогою об'єктів-заглушок. IoC-контейнер може витягувати класи без будь-якої настройки.

- Запити форм

Це перевірка запитів для ваших контролерів. Тобто повна обробка запиту.

- Eloquent

Eloquent ORM, це найпотужніша реалізація шаблону ActiveRecord в PHP. Крім CRUD-операцій, в ній є видалення, відносини, методи доступу і мутатори, мутатори дат, області запитів, спостерігачі моделі та ін.

- Цикл версій

Оновлення для Laravel виходять дуже часто. Через кожні шість місяців.

Ви отримуєте або версію з новими функціями, або версію з абсолютно новим номером. Версії з виправленнями помилок виходять кожні кілька місяців. Сам фреймворк розробляється дуже швидко.

- SSH завдання

У Laravel є досить простий спосіб для SSH-підключень до віддалених серверів і запуску команд, що дозволяє вам легко створювати завдання, які працюють на віддалених серверах.

- Flysystem

Flysystem - чудовий пакет для управління файловими системами. Це файлова система, яка використовує коннектори, за допомогою якої ви легко можете взаємодіяти з різними файловими системами в хмарі.

Якщо ви користувалися новітніми технологіями для фронтенду останні кілька років, то, ймовірно, ви використовували препроцесор для ваших файлів CSS і JavaScript. Але якщо немає, то Laravel про це подбає.

Вам потрібна максимально оптимізована онлайн-версія вашої програми. Це означає, що вам треба зменшувати і комбінувати ваші файли CSS і JavaScript.

Замість використання для цього будь-яких PHP-пакетів, Laravel пропонує пакет Elixir для виконавця завдань NodeJS Gulp. Тому ви можете використовувати всі ті переваги NodeJS і Gulp, які вам подобаються.

- compiled.php

Багато фреймворків завантажується з тисячею файлів і оголошень. Здебільшого, всі вони обробляються автозавантаженням Composer, але для найбільш використовуваних файлів Artisan може створити файл compiled.php, який завантажується при кожному запиті і об'єднує всі класи, які використовуються в вашому додатку.

Найбільш використовувані файли і класи компілюються в один-єдиний файл, який значно зменшує час завантаження програми.

Потім цей файл оптимізується виконуючою середовищем PHP, тому ви можете уникнути зниження продуктивності при завантаженні тисяч файлів в ваше робоче оточення.

- HHVM

Якщо ви захочете підвищити продуктивність вашого PHP-коду, ви можете змінити PHP-оточення на високооптимізоване середовище виконання для PHP від Facebook - HHVM.

Laravel - один з перших фреймворків, що підтримують HHVM.

- Cashier

Laravel Cashier забезпечує інтерфейс для послуг білінгових підписок Stripe. Він обробляє майже всі шаблонні коди білінгових підписок.

На додаток до основного управління підписками, Cashier може обробляти купони, підміну підписок, «кількість» підписок, періоди знижок і навіть генерувати PDF-файли рахунків.

Composer

- Composer - менеджер залежностей для PHP. Він не призначений конкретно для Laravel, але Laravel без нього не працює.
- Composer (з Packagist) містить тисячі PHP-пакетів від спільноти, більшість з яких мають мінімум залежностей і можуть бути просто і без проблем додані в вашу програму. Тому вам не потрібно винаходити велосипед, а фреймворк не заважатиме вам використовувати існуючі рішення.
- Composer обробляє автозагрузку PHP-класів, використовуючи будь-яку конфігурацію з карти класів, файлів, PSR-0 або PSR-4.
- Оскільки Composer використовує версійність, ви легко можете зафіксувати версії пакетів для вашої програми. Так ви будете впевнені, що всюди, де ви розвертаєте свій додаток, буде запускатися однаковий код.
- Почати працювати з Laravel дуже просто, достатньо встановити Composer і виконати:
- composer create-project laravel / laravel

PHP 5.4

- Laravel 5 вимагає PHP 5.4, а Laravel 4 вимагає PHP 5.3. І «вимагає», це означає, що Laravel повністю використовує всі нові можливості PHP, такі як функції-замикання, простори імен, типажі та інші.

- Laravel аж ніяк не є одним з фреймворків для PHP 4, адаптованих для використання класів PHP 5. Він розроблений в суворій відповідності з парадигмами ООП, і ви дійсно можете використовувати новітні можливості PHP.

- Те, що для Laravel потрібно PHP, також означає, що ваш код отримає покращену продуктивність нової середовища виконання PHP.

Dotenv

- Де ви зберігаєте секретну інформацію вашого застосування, таку як пароль від БД, логін для email та інші речі? Laravel 5 використовує добре відомий файл .env, який також використовують багато фреймворки для інших мов програмування. Це файл, в якому ви визначаєте пари ключ-значення для будь-якої вашої секретної інформації.

- Наприклад, ваш файл .env може виглядати так:

```
APP_ENV=production
DB_HOST=127.0.0.1
DB_DATABASE=laraveldb
DB_USERNAME=laravelapp
DB_PASSWORD=str0ngP4sw0rd
```

- І ви можете заповнити ваші файли налаштувань ось так:

```
'connections' => array(
    'mysql' => array(
        'driver'    => 'mysql',
        'host'      => getenv('DB_HOST'),
        'database'  => getenv('DB_DATABASE'),
        'username'  => getenv('DB_USERNAME'),
        'password'  => getenv('DB_PASSWORD'),
        'charset'   => 'utf8',
        'collation' => 'utf8_unicode_ci',
        'prefix'    => '',
    ),
),
```

- Ви можете створити цей файл локально і ще один на вашому робочому сервері і потім оновлювати ваш додаток, не турбуючись про БД та інші важливі секретні дані.

- Завантаження файлу .env автоматично обробляється Laravel за допомогою бібліотеки Dotenv.

PSR-4

- Laravel - перший фреймворк, що підтримує PSR-4. Прямо з коробки Composer автоматично завантажить всі класи з каталогу app, використовуючи стандарт автозавантаження PSR-4.

- Це означає, що у вас може бути один простір імен для вашого застосування, і ви можете структурувати його, як захочете і як вважаєте за логічним. Laravel не вимагає від вас розміщення певних файлів в певних папках.

- Це каталог зі стандартними програмами з простором імен App:

- app
  - Console
    - InspireCommand.php
  - Http
    - Controllers
      - Auth
        - AuthController.php
        - PasswordController.php
      - HomeController.php
    - Middleware
      - AuthMiddleware.php
      - BasicAuthMiddleware.php
      - CsrfMiddleware.php
      - GuestMiddleware.php
      - MaintenanceMiddleware.php
    - Requests
      - Auth
        - LoginRequest.php
        - RegisterRequest.php
  - Providers
    - AppServiceProvider.php
    - ArtisanServiceProvider.php
    - ErrorServiceProvider.php
    - EventServiceProvider.php
    - LogServiceProvider.php
    - RouteServiceProvider.php
  - User.php

- Як бачите, тут є деякі речі за замовчуванням, але ви можете розмістити класи там, де захочете, і це робиться дуже просто.

- Ви завжди можете змінити простір імен додатки за допомогою команди:

  SHphp artisan app: name YourNamespace

- Ця команда пройде по всіх ваших файлів і змінить в них оголошення простору імен.

IoC-контейнер - розширення ядра

- Контейнер зворотного управління в Laravel - потужний інструмент для управління залежностями класів. Впровадження залежностей ц спосіб зняття жорстко закодованих залежностей класу. Замість цього залежності впроваджуються під час виконання, забезпечуючи більшу гнучкість, так як реалізація залежностей може бути легко змінена.

- Ви можете використовувати IoC-контейнер в вашому додатку, щоб зробити все що впроваджуються і легко тестується за допомогою об'єктів-заглушок.

- IoC-контейнер може витягувати класи без будь-якої настройки. наприклад:

```
class FooBar {

  public function __construct(Baz $baz)
  {
    $this->baz = $baz;
  }

}
```

Як бачите, клас Baz автоматично впроваджений в клас FooBar. На практиці це буде використовуватися в контролерах ось так:

```
class OrderController extends BaseController {

  public function __construct(Order $orders)
  {
    $this->orders = $orders;
  }

  public function getIndex()
  {
    $all = $this->orders->all();

    return View::make('orders', compact('all'));
  }

}
```

- У цьому прикладі ви бачите, що клас Order впроваджений в OrderController без будь-якої додаткової настройки.

- Але в новому Laravel є дещо ще більш чудове, і називається воно - впровадження методів. Це впровадження залежностей на основі методів.

- Наприклад, цей клас може бути ще простіше:
- class OrderController extends BaseController {

```
  public function getIndex(Order $orders)
  {
    $all = $orders->all();

    return View::make('orders', compact('all'));
  }

}
```

- Ви можете додати стільки залежностей на основі класів або методів, скільки забажаєте, а IoC-контейнер витягне їх для вас.

- Оскільки все в Laravel внедряемо, ви легко можете підмінити що-небудь в ядрі фреймворку.

Запити форм

- У Laravel є ця чудова можливість, яка називається запитами форм. Це перевірка запитів для ваших контролерів. Це не просто перевірка даних, а повна обробка запиту.

- Запити форм комбінуються з функціональністю впровадження методів, щоб додати безшаблонний спосіб перевірки користувальницького введення. Наприклад, клас RegisterRequest:

```php
<?php namespace App\Http\Requests;

class RegisterRequest extends FormRequest {

  public function rules()
  {
    return [
      'email' => 'required | email | unique:users',
      'password' => 'required | confirmed | min:8',
    ];
  }

  public function authorize()
  {
    return true;
  }

}
```

- У цього запиту є правила для перевірки даних і функціональність авторизації, яка визначає, хто може використовувати цей запит. У цьому прикладі це може бути хто завгодно, тому будь-хто може зареєструватися, використовуючи поля email і password.

- Щоб використовувати цей запит, вам треба тільки натякнути на об'єкт запиту в вашому контролері, і ви отримаєте перевірку запиту дуже простим шляхом:

```
<?php namespace App\Http\Controllers\Auth;

use Illuminate\Routing\Controller;
use Illuminate\Contracts\Auth\Guard;
use App\Http\Requests\Auth\LoginRequest;
use App\Http\Requests\Auth\RegisterRequest;

class AuthController extends Controller {

  protected $auth;
```

```
public function __construct(Guard $auth)
 {
  $this->auth = $auth;
 }

 public function register(RegisterRequest $request)
 {
  // Форма регистрации прошла проверку, создать пользователя...
  $this->auth->login($user);

  return redirect('/');
 }
}
```

- Ви бачите, що в цьому контролері немає будь-якого сполучного коду або коду перевірки, все зроблено окремо і чисто.

- У запитів форм є функціональність перенаправлення для випадків помилок перевірки введення, повідомлень для уявлень і т.д.

Eloquent

- Eloquent ORM, включена в Laravel, - найбільш потужна реалізація шаблону ActiveRecord в PHP. Крім звичайних CRUD-операцій в ній є м'яке видалення, області запитів, відносини, методи доступу і мутатори, мутатори дат, спостерігачі моделі і багато іншого.

- Вам навіть не потрібен Laravel, щоб її використовувати. Ви легко можете додати шар БД Laravel з Eloquent в ваш PHP-проект, зроблений не так на Laravel. Перейдіть по посиланню database, щоб налаштувати її в вашому проекті.

- Не кажучи вже про сотні розширень Eloquent, таких як вкладений набір Baum, Translatable, перевірка моделей Ardent, Database Backup Manager, MongoDB і багатьох інших.

Цикл версій

- У Laravel дуже передбачуваний цикл версій. Через кожні шість місяців можна чекати чергової версії. Laravel виходить через один місяць після виходу Symfony.

- Ви отримуєте або версію з новими функціями, або версію з абсолютно новим номером. Версії з виправленнями помилок виходять кожні кілька місяців. Це добре, тому що фреймворк розробляється дуже швидко, і ви можете розраховувати на отримання поліпшень безпеки та інших поліпшень так швидко, як тільки можливо.

- На основні версії особливо легко переходити, і у вас, ймовірно, не виникне питань при оновленні. Але оскільки Laravel є відмінною основою для чого б то не було, ви можете навіть залишатися на одній версії протягом всього року і виявите, що для цієї версії, як і раніше виходять виправлення і поліпшення.

SSH задачі

- У Laravel є простий спосіб для SSH-підключень до віддалених серверів і запуску команд, що дозволяє вам легко створювати завдання, які працюють на віддалених серверах. За допомогою невеликої настройки ви зможете запускати ваші завдання ось так:

```
SSH::into('production')->run(array(
  'cd /var/www',
  'git pull origin master',
));
```

- Або ви можете використовувати підключення до іншого сервера:

```
SSH::into('staging')->run(array(
  'cd /var/www',
  'git pull origin master',
));
```

- Ви також можете завантажувати файли:

```
SSH::into('staging')->get($remotePath, $localPath);
$contents = SSH::into('staging')->getString($remotePath);
```

- Або загружати:

- ```
  SSH::into('staging')->put($localFile, $remotePath);
  SSH::into('staging')->putString($remotePath, 'Foo');
  ```

- Отже, ви побачили, що ви можете створювати за допомогою цього інструменту. Ви можете автоматизувати кілька завдань, які потрібні вам для розгортання або обслуговування вашого сервера, або для чогось ще.
  ```

Flysystem

- Flysystem - чудовий пакет з Ліги видатних пакетів для управління файловими системами. Це файлова система, яка використовує коннектори, за допомогою якої ви легко можете взаємодіяти з local, awss3, dropbox, rackspace і іншими файловими системами в хмарі.

- Ось так просто:

- ```
  // Запись файлів
  File::put('filename.txt', 'contents');

  // Зчитування файлів
  $contents = File::get('filename.txt');
  ```
  ```

```
// Створення каталогу
File::makeDirectory('nested/directory');

// Видалення каталогу
File::deleteDirectory('path/to/directory');
```

- І ви можете використовувати цей же API для будь-якої файлової системи, яку хочете використовувати. Flysystem поставляється вбудованим в Laravel, і ви можете використовувати клас File.

Elixir

- Якщо ви користувалися новітніми технологіями для фронтенда останні кілька років, то, ймовірно, ви використовували препроцесор для ваших файлів CSS і JavaScript. Але якщо немає, то Laravel про це подбає.

- Вам потрібна максимально оптимізована онлайн-версія вашої програми. Це означає, що вам треба зменшувати і комбінувати ваші файли CSS і JavaScript.

- Замість використання для цього будь-яких PHP-пакетів Laravel пропонує пакет Elixir для виконавця завдань NodeJS Gulp. Тому ви можете використовувати всі ті переваги NodeJS і Gulp, які вам подобаються.

- Ви запускаєте Gulp-завдання watch, і Elixir подбає про все для вас. Якщо ви використовуєте LESS і CoffeeScript, то ваші налаштування можуть бути такими:

```
var elixir = require('laravel-elixir');
elixir(function(mix) {
  mix.less()
     .coffee()
     .routes()
     .events()
     .phpUnit();
});
```

- Ця конфігурація скомпілює ваші файли less і coffee, а також просканує маршрути і події в вашому додатку. Ще вона буде запускати юніт-тести після кожної зміни PHP-файлів.

- Це так просто налаштувати. Вам треба тільки помістити ваші файли less і coffee в папку / resources / assets. Для less - / resources / assets / less, а для coffee - / resources / assets / coffee. Тепер все налаштовано.

- Далі ви можете посилатися на створені файли в ваших уявленнях і створювати файли пізніше для робочого оточення. Насправді Elixir надає навіть більше можливостей, ніж показано в прикладі.

compiled.php

- Будь-який повнофункціональний фреймворк завантажується з тисячею файлів і оголошень. Здебільшого всі вони обробляються автозавантаженням Composer, але для найбільш використовуваних файлів Artisan може створити файл compiled.php, який завантажується при кожному запиті і об'єднує всі класи, які використовуються в вашому додатку.

- Найбільш використовувані файли і класи компілюються в один єдиний файл, який значно зменшує час завантаження програми. Якщо у вас є файли, які потрібно завантажувати при кожному запиті, ви також можете вказати, які файли компілювати в compiled.php.

- Потім цей файл оптимізується виконуючим середовищем PHP, тому ви можете уникнути зниження продуктивності при завантаженні тисяч файлів в ваше робоче оточення.

HHVM

- Коли ви захочете підвищити продуктивність вашого PHP-коду, ви можете змінити PHP-оточення на високооптимізоване оточення виконання для PHP від Facebook - HHVM.

- Laravel - один з перших фреймворків, що підтримують HHVM, і він проходить 100% юніт-тестів на HHVM. Тому ви можете бути впевнені, що легко можете переключити ваше додаток з PHP на HHVM і отримати ще більше продуктивності.

Homestead

- Laravel Homestead - офіційна упакована Vagrant- "коробка", що надає вам дивовижне середовище розробки, не вимагаючи установки PHP, HHVM, веб-сервера і будь-якого іншого ПО на вашу локальну машину.

- Це віртуальна машина з встановленим ПО, в яке входять:
- - Ubuntu 14.04
- - PHP 5.6
- - HHVM
- - Nginx
- - MySQL
- - Postgres
- - Node (з Bower, Grunt і Gulp)
- - Redis
- - Memcached
- - Beanstalkd
- - Laravel Envoy
- - Fabric + розширення HipChat

- Ви просто запускаєте vagrant up, заходите на locahost: 8000 і бачите ваше запущене застосування. Ваш код локально «розшарено» для віртуальної машини, і вам взагалі не треба встановлювати все перераховане ПО, а потрібен лише Laravel і ваш редактор коду.

- Пізніше, коли ви захочете створити VPS-сервер, ви можете використовувати проект Laravel Settler для створення точно такого ж сервера, як ваше оточення Homestead.

- Це просто чудово і економить так багато часу, позбавляючи вас від тужних налаштувань локального оточення для розробки.

Проміжне ПО Stack

- Laravel використовує реалізацію Symfony HttpKernel від StackPHP. Це означає, ми можемо додати наше власне проміжне ПО в HTTP-шар!

- Проміжне ПО запитує обробники, за допомогою яких ви можете додати функціональність в HTTP-запити - до, після і навіть змінюючи запити і відповіді.

- Наприклад, подивимося на MaintenanceMiddleware:

-
```
<?php namespace App\Http\Middleware;

use Closure;
use Illuminate\Http\Response;
use Illuminate\Contracts\Routing\Middleware;
use Illuminate\Contracts\Foundation\Application;
```

```
class MaintenanceMiddleware implements Middleware {

    /**
     * Реализация приложения.
     *
     * @var Application
     */
    protected $app;

    /**
     * Создание нового экземпляра фильтра.
     *
     * @param  Application  $app
     * @return void
     */
    public function __construct(Application $app)
    {
        $this->app = $app;
    }

    /**
     * Обработка входящего запроса.
     *
     * @param  \Illuminate\Http\Request  $request
     * @param  \Closure  $next
     * @return mixed
     */
    public function handle($request, Closure $next)
    {
        if ($this->app->isDownForMaintenance())
        {
            return new Response('Be right back!', 503);
        }

        return $next($request);
    }

}
```

- Ви бачите, що функціональність службової перевірки додана в кожен запит всередині методу handle. Бачите цей рядок $ next ($ request)? Це те, як запит передається наступного класу проміжного програмного забезпечення і в кінці самому фреймворку.

- Ви можете використовувати проміжне ПО для додавання будь-якої необхідної функціональності в ваш HTTP-шар.

- За замовчуванням в Laravel є AuthMiddleware, GuestMiddleware, MaintenanceMiddleware, CsrfMiddleware, і ви легко можете додати ваше власне проміжне ПО або використовувати одне з доступних тут, таке як HttpCache, Geoip, CORS, OAuth, Turbolinks і інше.

Cashier

- Laravel Cashier забезпечує виразний, гнучкий інтерфейс для послуг білінгових підписок Stripe. Він обробляє майже всі шаблонні коди білінгових підписок, які ви боїтеся написати.

- На додаток до основного управління підписками Cashier може обробляти купони, підміну підписок, «кількість» підписок, періоди знижок, і навіть генерувати PDF-файли рахунків.

- Це офіційний пакет від розробників Laravel, що не включений за замовчуванням. Подивіться, як просто використовувати Cashier для підписки користувача:

```
$user = User::find(1);
$user->subscription('monthly')->create($creditCardToken);

// з купоном
$user->subscription('monthly')
    ->withCoupon('code')
    ->create($creditCardToken);

// конец пробного періоду
$user->trial_ends_at = Carbon::now()->addDays(14);
$user->save();

// Відміна
$user->subscription()->cancel();

// Підсумок
$user->subscription('monthly')->resume($creditCardToken);

// Перевірка статусу
if ($user->subscribed())
{
  //
}
```

```php
// На випробному періоді
if ($user->onTrial())
{
  //
}

// Відмінений
if ($user->cancelled())
{
  //
}

// В період скидок
if ($user->onGracePeriod())
{
  //
}

// Був колись підписаний
if ($user->everSubscribed())
{
  //
}

// Проверка тарифу користувача
if ($user->onPlan('monthly'))
{
  //
}
```

- Ви можете використовувати рахунки і багато інших функцій Stripe.

Rocketeer

- Rocketeer - швидкий і легко розгортається інструмент для сучасних розробників. Він натхненний Laravel і може бути використаний в Laravel, а також в будь-якому PHP-проекті.
- Він може розгорнути будь-який проект від маленького HTML / CSS веб-сайту до великого PHP-додатки, або будь-який додаток на будь-якій мові програмування.
- За допомогою Rocketeer ви легко можете розгортати останнім, поточний або проміжне оточення і відкочуватися до старих версій при необхідності.
- Після невеликої настройки запустити Rocketeer так само легко, як запустити artisan і одну з цих команд:

- deploy

deploy:check	Перевірка готовності сервера отримати додаток
deploy:cleanup	Видалити старі версії з сервера
deploy:current	Вивести поточну версію
deploy:deploy	Розгорнути веб-сайт
deploy:flush	Очистити кеш повноважень Rocketeer
deploy:ignite	Створити конфігурацію Rocketeer
deploy:plugin-config	Опублікувати конфігурацію плагіна
deploy:plugin-install	Встановити плагін
deploy:plugin-list	Перегляд списку поточних включених плагінів
deploy:rollback	Відкат до попередньої версії, або до конкретної
deploy:setup	Встановити віддалений сервер для розгортання
deploy:strategies	Перегляд списку доступних варіантів для кожної стратегії
deploy:teardown	Видалити віддалене додаток і існуючий кеш
deploy:test	Запустити тести на сервері і вивести результати
deploy:update	Оновити віддалений сервер не створюючи нову версію

- Зручно використовувати такий надійний і продуманий процес розгортання.

Об'єднуючи PHP

- Laravel дійсно об'єднує PHP-співтовариство, використовуючи більше 23 пакетів від всієї PHP-спільноти. Використання «кращих з кращих» PHP-пакетів сприяє більш тісній взаємодії Laravel і всієї PHP-спільноти. Серед включених в Laravel пакетів присутні:

- Dotenv: PHP-версія оригінального dotenv з Ruby, завантажує змінні середовища.

- Carbon: значна бібліотека для роботи з датами від Брайана Несбіта (Brian Nesbitt).

- Predis: надійний Redis-клієнт, створений Даніелем Алесандро (Daniele Alessandri).
- Phenstalk: повнофункціональний PHP-клієнт для черги Beanstalked.
- SuperClosure: написана Джеремі Ліндблумом (Jeremy Lindblom), ця потужна бібліотека дозволяє вам серіалізувати і десеріалізувати функції-замикання в PHP.
- Whoops: виводить красиві сторінки помилок і трасування стека поки Laravel в режимі розробки.
- Monolog: стандарт де-факто серед PHP-бібліотек для ведення логів.

- Boris: дійсно відмінний PHP REPL, який підтримує чудову консольную команду "tinker".
- PasswordCompat: забезпечує захищене хешування Bcrypt, яке використовується в Laravel за замовчуванням.
- Symfony HttpFoundation: абсолютно надійна HTTP-абстракція. Протестована і перевірена в багатьох великих реальних додатках.
- Symfony Routing: цей пакет забезпечує компіляцію маршрутів Laravel в регулярні вирази.

- Symfony HttpKernel: містить HttpKernelInterface, який використовується як абстракція нижнього рівня в додатках Laravel.
- Symfony BrowserKit: відмінне функціональне тестування.
- StackPHP: цей проект описує структуру для побудови багаторазового незалежного від фреймворків проміжного програмного забезпечення на рівні HTTP-шару.

Laracasts - Джеффрі Вей (Jeffery Way)

- Laracasts - це ресурс не тільки для Laravel-розробників, але і для сучасних PHP-розробників в цілому. Він надає більше 300 відео для вирішення будь-якої задачі або реалізації практично чого завгодно за допомогою Laravel.

- Майже все, що згадано в цій статті, можна дізнатися на Laracasts. Це дивовижний навчальний ресурс, який створив Джеффрі Вей. Джеффрі Вей - дуже шанований учасник PHP-спільноти і відмінний учитель.

- З Laracasts співтовариство Laravel дійсно виділяється з екосистеми PHP, так як немає нічого подібного для інших PHP-фреймворків.

Liferaft

- Створення запитів на GitHub - кращий спосіб повідомити про проблему і отримати рішення. Але що ви робите, коли не можете пояснити свою проблему? Що якщо ви стикаєтеся з цією проблемою тільки в коді вашої програми?

- Liferaft (рятувальний пліт - прим. Пер.) - Інструмент з інтерфейсом командного рядка, який встановлюється з Composer, який забезпечує кращий спосіб сприяти Laravel чи повідомляти про проблеми.

- Ви просто запускаєте одну з команд Liferaft, і вона створює додаток Laravel. І ви просто ставите свій код, який відтворює вашу проблему, і запускаєте іншу команду Liferaft для створення запиту на Github.

- наприклад:

```
liferaft new my-bug-fix
```

- І програвайте проблему в цьому додатку. Єдине, що від вас вимагається, це заповнити файл liferaft.md описом вашої проблеми, і ви можете надати юніт-тест, який показує, як це повинно працювати. І потім:

```
liferaft throw
```

- Тоді розробник зможе запустити вашу програму і побачити вашу проблему. Це вирішує безліч проблем і займає набагато менше часу у розробників фреймворку на відповідь і рішення вашої проблеми.

Socialite

- У Laravel 5.0 з'явився чудовий інструмент для авторизації через соціальні мережі - Socialite, але на офсайті наведено приклад налаштування тільки для GitHub.

- Для початку нам необхідно встановити даний пакет

  $ composer require laravel/socialite

- Після установки Socialite зареєструйте сервіс-провайдер Laravel \ Socialite \ SocialiteServiceProvider в файлі конфігурації config / app.php
- 'providers' => [... Laravel\Socialite\SocialiteServiceProvider::class,],
- Також додайте фасад Socialite в масив aliases конфігурації:
- Установка на цьому закінчена. Приступимо до конфігурації.

- В прикладах будуть показані чотири сервіси (GitHub, Google, Facebook і Twitter)

- Для початку створимо два маршрути в app \ Http \ routes.php

```
Route::get(
    '/socialite/{provider}',
    [
        'as' => 'socialite.auth',
        function ( $provider ) {
            return \Socialite::driver( $provider )->redirect();
        }
    ]
);

Route::get('/socialite/{provider}/callback', function ($provider) {
  $user = \Socialite::driver($provider)->user();
  dd($user);
});
```

- Перший маршрут призначений для переходу на сервіс для авторизації. Другий приймає перехід з сервісу і відображає дані користувача при позитивному результаті авторизації.

Спільнота і документація

- Сьогодні кожен фреймворк має велике і дружнє співтовариство, і Laravel - не виняток. Люди, що підтримують і розробляють Laravel-пакети, дуже доброзичливі, і ви легко можете брати участь у створенні будь-якого пакета, і ваші правки будуть прийняті дуже скоро.

- Це сприяє розширенню спільноти, і вам варто знати, що Laravel був внесений до списку найпопулярніших PHP-фреймворків в 2013 році, випередивши Phalcon, Symfony2 і CodeIgniter. А в серпні 2014 року Laravel став найпопулярнішим PHP-проектом на GitHub.

Висновок

- Laravel є найпопулярнішим фреймворком на сьогоднішній день і продовжує стрімко розвиватися. Зараз Laravel - це величезна екосистема, що включає хостинг та платформу для розгортання додатків. Сьогодні найбільше проектів, що розробляються за допомогою фреймворків, створюються саме з використанням Laravel. При розміщенні вакансій PHP-розробників, веб-студії все частіше включать знання фреймворка Laravel як обов'язково умова. Тому, якщо ви до сих пір не знайомі з цим чудовим фреймворком, то зараз саме час почати його вивчення.

Кінець