web разработка

MVC

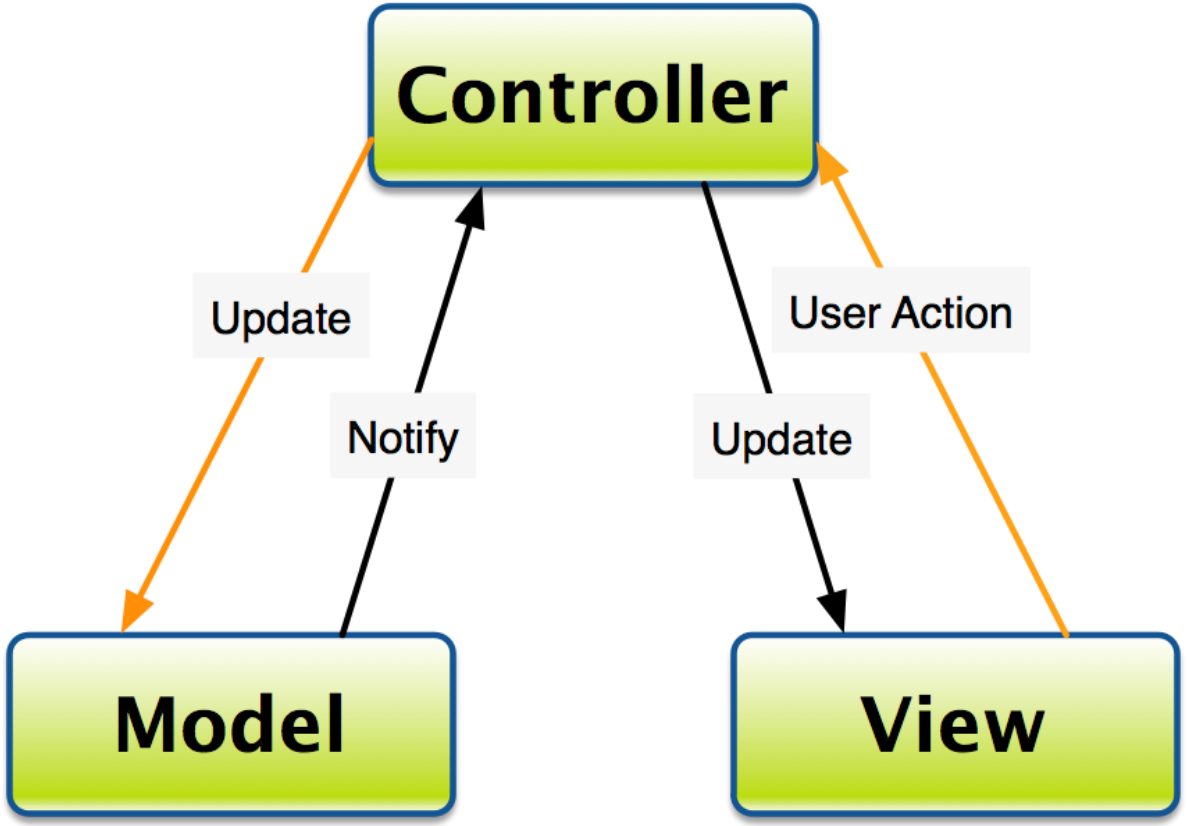
Съдържание

- MVC & Frameworks
- Laravel
- Composer
- Инсталация на Ларавел
- Namespaces, Traits
- Документация на Ларавел

MVC

MVC

Model–view–controller (**MVC**)
е архитектурен шаблон за дизайн
в програмирането,
основан на разделянето на
бизнес логиката
от графичния интерфейс
и данните в дадено приложение.



MVC

Предимства и недостатъци на MVC

+ Успоредна разработка - позволява група разработчици да разработват едновременно дадено приложение.
+ MVC позволява логическо групиране.
+ Всяка една част на MVC позволява да бъда планирана, осъществена и променяна независимо от останалите части на приложението.

- Оганизацията на файловете в приложението е усложнена. Но от друга страна, тази организация не се променя за различните приложения, ползващи един и същ MVC framework.
- Познаването на множество технологии е задължително за разработчиците, използващи MVC.

Framework

Framework

Защо фреймуърк?

Готовите библиотеки и компоненти представляват съвкупност от кодове, с ясно дефинирано предназначение. Те са разработени, поддържат се и се доразвиват от хора, които ги познават напълно и носят отговорност за безпроблемната им работа.

Laravel, Symfony, Silex, Lumen, Slim и др. Framework представляват колекция от такива компоненти, обединени чрез специфична за конкретния фреймуърк свързваща част - конфигурационни файлове, service providers, точно определна структура на директориите и др. Така че, ползата от използването на фреймуърк, най-просто казано е в това, че някой друг е преценил какви компоненти да групира и то по такъв начин, че да работят максимално добре заедно.

Laravel

Laravel

Първата експериментална версия на Laravel е пусната през юни 2011.

Началното развитие на Laravel е много бързо и съответно

Laravel 2 и 3 са публикувани през ноември 2011 и февруари 2012.

Laravel


Laravel is the strongest contender in the PHP ecosystem simply because it includes the features needed to build modern web applications.
Taylor Otwell,
Creator of Laravel, Forge, Envoyer

Laravel

Laravel 4

Авторът пренаписва всичко отначало.

Към този момент Composer/ PHP package manager/започва да се превръща в стандарт за индустрията и Тейлър преценява, че си струва да пренапише Ларавел като колекция от компоненти, свързани и работещи заедно благодарение на Composer.

Laravel

Laravel 5

Laravel 4.3 е бил планиран да излезе през ноември 2014 г., но в процеса на разработката му, направените промени са толкова значителни, че е издадена съвсем нова - Ларавел 5 версия през февруари 2015 г.

Laravel 5.5 - излиза на 30 август 2017 /!изисква php 7/

Ново в Ларавел 5.5

Laravel

И така … какво прави Ларавел толкова специален?

Кое е го отделя от останалите?

Защо е нужно да има повече от един PHP фреймуърк?

Laravel

Философията на Ларавел

Прегледайте маркетинг материалите на Ларавел, неговите README файлове.

Ще ви направят впечатление свързаните със светлина имена, които са навсякъде в Ларавел - “Illuminate”, “Spark”...

Laravel

Две от най-популяризираните ценности на Ларавел са

- Скоростта на работа и
- Удовлетворението от работата за самия разработчик

Laravel

“...Понякога прекарвам ужасно дълго време /часове/ агонизирайки в опитите си да накарам кода, който пиша да изглежда приятно. Само заради удоволствието да преглеждам самия код... “

“... Трябва да е възможно по-лесно и по-бързо да превръщаме идеите си в реалност, като се премахнат ненужните бариери пред създаването на страхотни продукти…”

Тейлър Отуел,

създател на Ларавел

Laravel

Ларавел е посветен на екипността и даването на възможности на разработчиците.

Целта му е да се създава ясен, прост и красив код. Притежава функционалности, които помагат - изучават се бързо, така че разработчиците могат да стартират бързо проектите си, да пишат прост и ясен, код който да остава във времето.

Laravel

В документацията може да намерите изречението

“Happy developers make the best code”

“Developer happiness from download to deploy”

- неофициално мото на Ларавел за известно време.

Laravel

Как прави живота на разработчиците по-лесен /и щастлив/?

Първо, Laravel е фреймуърк за бързо разработване на приложения. Той се изучава лесно и е съсредоточен върху минимизиране на стъпките между стартирането на разработването на приложение и публикуването му.

Всички от най-често използваните задачи при разработване на уебприложение - от работата с базата данни, до проверка на потребителите, изпращане на имейли, опашки, кеширане - са облекчени чрез компонентите, които Ларавел съдържа.

Laravel

Но компонентите на Ларавел, не са просто много добри сами по себе си. Те предлагат консистента и предвидима структура за целия фреймуърк. Това означава, че когато пробвате нещо ново в Ларавел, много е вероятно да накрая да кажете

“... хм, и то просто работи!?....”

Laravel

Това не свършва с пакета Ларавел, който ще свалите и инсталирате.

Ларавел разполага и с цяла екосистема от инструменти за разработване и пускане в експлоатация на приложение /Homestead, Vallet/ и допълнителни библиотеки /packages/.

Laravel

Ларавел се фокусира върху конвенцията за сметка на конфигурацията.

Или ако имате намерение да използвате конфигурацията по подразбиране на Ларавел, ще ви се наложи да свършите много малко допълнителна работа в сравнение с други frameworks, които изискват от вас да декларирате всички настройки дори и ако използвате препоръчителните.

В сравнение с повечето PHP frameworks, проектите разработвани на Ларавел изискват по-малко време.

Laravel

Ларавел се фокусира върху простотата. Може да използвате всички най-сложни шаблони за проектиране, които познавате, ако желаете. Но за разлика от другите frameworks, които ще ви препоръчват да ги използвате във всеки проект, Laravel, документацията и общността на Ларавел, ще ви препоръчат да започнете с

възможно най-простата имплементация.

Това позволява на разработчиците да създадат възможно **най-простото приложение, разрешаващо даден проблем.**

Laravel

Авторът на Ларавел и неговите последователи са свързани и се вдъхновяват от Ruby и Rails и езиците, използващи функционално програмиране.

Съществува силно съвременно течение в PHP, което клони към усложняване и утежняване на разработките, поддържайки тези страни от PHP, наподобяващи Java.

Laravel се стреми да бъде от другата страна - той подкрепя

експресивните, динамични и опростени практики

и страни на езика за програмиране.

Laravel

Общността на Ларавел

Един от отличителните елементи на Ларавел, допринасящ за неговото израстване и успех, е добронамерената обучаваща общност, която го обгражда.

Ларавел разполага с богата и жизнена общност от хора, които го подкрепят от първия ден и досега.

И това не е случайно ...

Laravel

… From the very beginning of Laravel, I've had this idea that all people want to feel like they are part of something. It's a natural human instinct to want to belong and be accepted into a group of other like-minded people. So, by injecting personality into a web framework and being really active with the community, that type of feeling can grow in the community.

—Taylor Otwell,Product and Support interview

Composer

Composer

Инсталация

Composer е инструмент, управляващ зависимостите в PHP. Той ви дава възможност да декларирате библиотеките, които използвате във вашия проект и се грижи за тях - инсталира ги, обновява ги, вместо вас.

composer.jason е файлът, в който за изброени нужните за вашия проект библиотеки.

Composer

Управляване на зависимостите

Composer се грижи за т.нар. Пакети и библиотеки, но за всеки отделен проект, като ги инсталира в една директория /например vendor/. По подразбиране не инсталира нищо глобално, освен ако не посочите командата global.

Идеята не е нова и Composer я наследява от npm на node и bundler на ruby.

Представете си, че

1. Имате проект, за който са нужни няколко библиотеки.
2. За някои от библиотеките, са нужни други библиотеки.

Composer:

1. Ви дава възможност да декларирате библиотеките, от които зависите.
2. Намира кои версии на кои пакети може и трябва да инсталира и ги инсталира /което означава, че ги сваля и инсталира във вашия проект/.

Инсталиране на Ларавел

Инсталиране на Ларавел

Изисквания към сървъра

PHP >= 5.6.4
OpenSSL PHP Extension
PDO PHP Extension
Mbstring PHP Extension
Tokenizer PHP Extension
XML PHP Extension

Инсталиране на Ларавел

Инсталиране на Ларавел - документация

Команда за инсталиране /трябва да сме в папката където ще създаваме проекта/

composer create-project --prefer-dist laravel/laravel името-на-проекта

Команда за инсталиране на конкретна версия

composer create-project laravel/laravel=версия-на-ларавел your-project-name --prefer-dist

Стартираме проекта в браузъра чрез

Пътя-до-проекта/public

Namespace

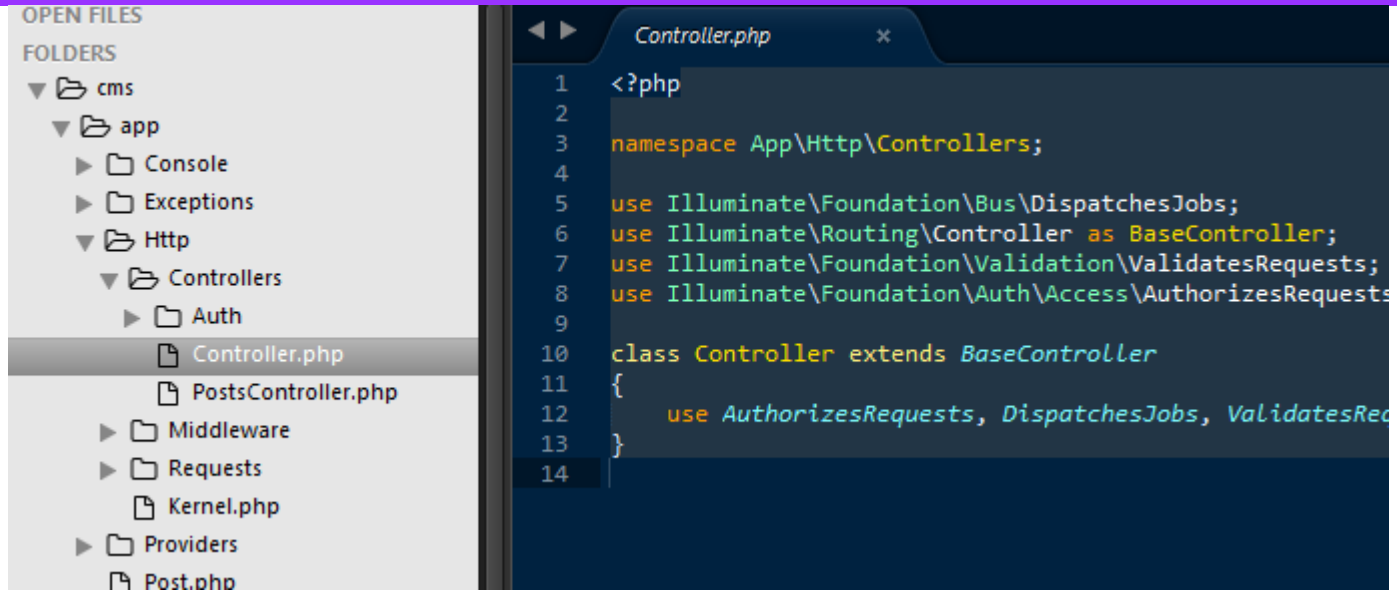
Namespace

Декларираме ги чрез ключовата дума *namespace*.

Файл съдържащ namespace, трябва да го декларира в началото на файла, преди всякакъв друг код - ведната след отварящия php таг.

Наподобяват файлова йерархия.
Използва се за облекчено извикване на класове, методи на класове и др.

Namespace



Обърнете внимание на пътя до файла в примера и връзката му с дефинирания namespace

Namespace

use ….

```php
use Illuminate\Foundation\Bus\DispatchesJobs;
use Illuminate\Routing\Controller as BaseController;
use Illuminate\Foundation\Validation\ValidatesRequests;
use Illuminate\Foundation\Auth\Access\AuthorizesRequests;

class Controller extends BaseController
{
    use AuthorizesRequests, DispatchesJobs, ValidatesRequests;
}
```

Декларираме кой неймспейс използваме в класа.
Обръщаме се към него в кода по последната дума в неймспейса
use Illuminate\Foundation\Bus\DispatchesJobs;
обръщаме се с DispatchesJobs

За да избегнем двусмислие използваме следния вариант на декларация
use Illuminate\Routing\Controller as BaseController;
обръщаме се с BaseController

Trait

Trait

Trait е създаден с намаренетието да се избегнат някои ограничения вследствие на това, че един клас може да наследява само един клас.

Дава възможност да се преизползват набор от методи в няколко независими един от друг класове, намиращи се в различни йерархии от класове.

Trait прилича на клас, но само дотолкова, че групира няколко метода, превръщайки ги в определена функционалност.

В него няма декларирани свойства.

Не може да създавате обекти от един трейт.

Trait

```php
class Controller extends BaseController
{
    use AuthorizesRequests, DispatchesJobs, Validates
}
```

Project Preferences Help

Controller.php | AuthorizesRequests.php

```php
<?php

namespace Illuminate\Foundation\Auth\Access;

use Illuminate\Contracts\Auth\Access\Gate;

trait AuthorizesRequests
{
    /**
     * Authorize a given action for the current user.
     *
     * @param  mixed  $ability
     * @param  mixed|array  $arguments
     * @return \Illuminate\Auth\Access\Response
     *
     * @throws \Illuminate\Auth\Access\AuthorizationException
     */
    public function authorize($ability, $arguments = [])
    {
        list($ability, $arguments) = $this->parseAbilityAndArgu
```

Документация на Ларавел

Документация на Ларавел

https://laravel.com/docs/

При използване на документацията, трябва да посочите за коя версия на Ларавел търсите информация.

Ресурси

ресурси

https://laracasts.com/

Edwin Diaz - PHP with Laravel, Become a master in Laravel