

Laravel

Что такое Laravel?

Laravel

Бесплатный PHP фреймворк с открытым исходным кодом

Создан Тейлором Отвеллом в 2011 году

Использует паттерн MVC

На появление Laravel повлияли остальные PHP фреймворки

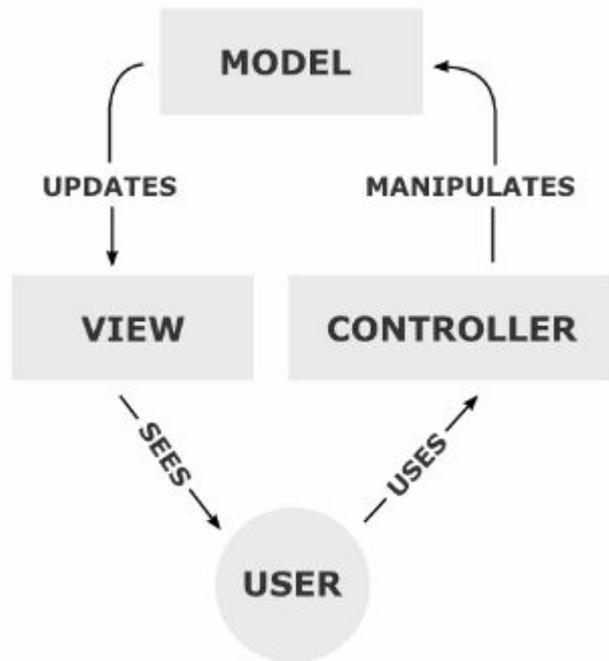
В качестве основы Laravel выступают компоненты другого фреймворка -
Symfony

Laravel плюсы и минусы

- + Достаточно неплохая и понятная документация
- + Мощная экосистема: различные курсы, конференции, обучающие материалы
- + Гибкая система маршрутизации
- + Удобный механизм обработки ошибок и исключений
- + Встроенные механизмы аутентификации и авторизации пользователей
- + Много синтаксического сахара
- Много синтаксического сахара
- Нарушение обратной совместимости между версиями фреймворка
- Документация и комьюнити на английском

Что такое MVC?

Model View Controller (Модель-Представление-Контроллер) — схема разделения данных приложения, и управляющей логики на три отдельных компонента: модель, представление и контроллер — таким образом, что модификация каждого компонента может осуществляться независимо



Model

Логика манипулирования данными

Взаимодействия с БД (SELECT, INSERT, UPDATE, DELETE)

Предоставляет данные и реагирует на команды контроллера, изменяя свое состояние

View

Отвечает за отображение данных модели пользователю, реагируя на изменения модели

Обычно содержит HTML & CSS

Общается с контроллером

Используются шаблонизаторы

Controller

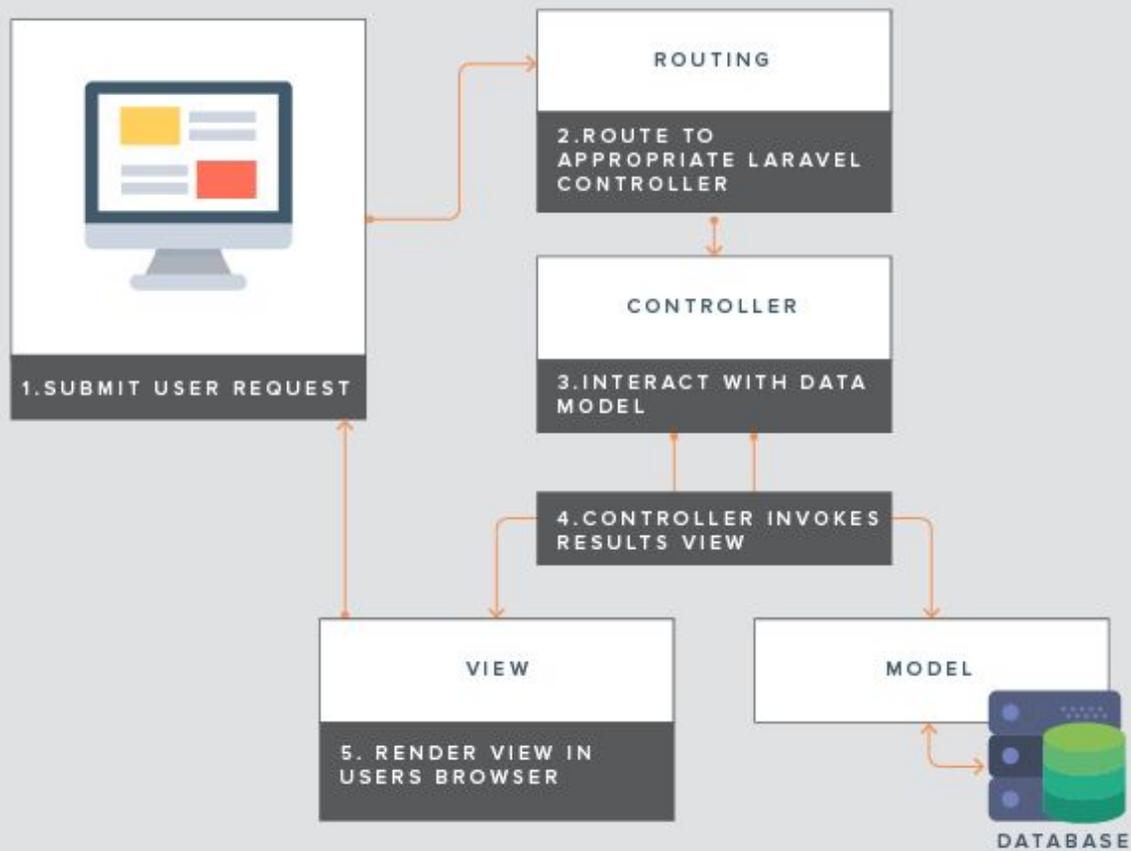
Получает данные

Обрабатывает запросы

Получает данные из модели

Передаёт данные в представление

MVC LARAVEL ARCHITECTURE



Установка и настройка

Требования

PHP >= 8.0

BCMath PHP Extension

Ctype PHP Extension

JSON PHP Extension

Mbstring PHP Extension

OpenSSL PHP Extension

PDO PHP Extension

Tokenizer PHP Extension

XML PHP Extension

Установка

```
composer create-project laravel/laravel example-app
```

```
cd example-app
```

```
php artisan serve
```

Composer — это пакетный менеджер уровня приложений для языка программирования PHP, который предоставляет средства по управлению зависимостями в PHP-приложении.



Установка в рамках инфраструктуры демоэкзамена

1. Копируем архив с фреймворком из директории public в необходимый модуль;
2. Распаковываем архив;
3. Настраиваем редиректы с помощью файла .htaccess;
4. Настраиваем подключение к БД в файле .env.

.htaccess

RewriteEngine On

RewriteRule ^(.*)\$ public/\$1

.env

...

DB_CONNECTION=mysql

DB_HOST=127.0.0.1

DB_PORT=3306

DB_DATABASE=login_m1

DB_USERNAME=login

DB_PASSWORD=password

...

Структура проекта

/app (содержит код ядра приложения)

/bootstrap (содержит файлы, которые загружают фреймворк и настраивают автозагрузку)

/config (содержит все конфигурационные файлы)

/database (содержит миграции и классы для наполнения начальными данными БД)

/public (содержит файл index.php, который является входной точкой для всех запросов, поступающих в ваше приложение)

/resources (содержит ваши представления, а также сырые, некомпиллированные ресурсы)

/routes (содержит все определения маршрутов вашего приложения)

/storage (содержит скомпилированные Blade-шаблоны, файл-сессии, кэши файлов и другие файлы, создаваемые фреймворком)

/tests (содержит автотесты)

/vendor (содержит Composer-зависимости)

Каталог app

/Console (содержит все дополнительные Artisan-команды)

/Exceptions (содержит обработчик исключений вашего приложения)

/Http (содержит пользовательские контроллеры, посредники и запросы форм)

/Models (содержит пользовательские модели)

/Providers (содержит все сервис-провайдеры приложения)

Artisan

Artisan - интерфейс командной строки для Laravel.

Команды экономят время.

Рекомендуется генерация файлов с помощью Artisan.

```
jurapro@jurapro:/opt/lampp/htdocs/example-app$ php artisan
```

```
Laravel Framework 8.58.0
```

Usage:

```
command [options] [arguments]
```

Options:

<code>-h, --help</code>	Display help for the given command. When no command is given display help for the <code>list</code> command
<code>-q, --quiet</code>	Do not output any message

Полезные команды Artisan

- ❑ `serve`
- ❑ `key:generate`
- ❑ `migrate`
- ❑ `migrate:fresh`
- ❑ `migrate:rollback`
- ❑ `make:model`
- ❑ `make:controller`
- ❑ `make:migration`
- ❑ `route:list`
- ❑ `storage:link`
- ❑ `tinker`