* Clone your project
* Go to the folder application using cd command on your cmd or terminal
* Run composer install on your cmd or terminal
* Copy .env.example file to .env on the root folder. You can type copy .env.example .env if using command prompt Windows or cp .env.example .env if using terminal, Ubuntu
* Open your .env file and change the database name (DB\_DATABASE) to whatever you have, username (DB\_USERNAME) and password (DB\_PASSWORD) field correspond to your configuration.
* Run php artisan key:generate
* Run php artisan migrate
* Run php artisan serve
* Go to http://localhost:8000/

**Подключение**

-Устанавливаем Composer

-Через командную строку устанавливаем Laravel(как указано на сайте Laravel)

-Устанавливаем наш новый проект Laravel в C:\OSPanel\domains(если установлен Open Server)

-В настройках Open Server и в разделе "Домены" настроим новый домен для нашего нового проекта. Домашний каталог у Laravel (там где лежит файл index.php) находится в каталоге "public", и именно его мы должны указать в качестве папки домена. Имя домена "mysite.loc" может быть любое



**Функции(хэлперы)**

**protected $table='имя таблицы';** – в Model указывать имя то таблицы с которой связана Model

**dd(переменная)** – показывает значение и останавливает выполнение кода

**dump(переменная)** – показывает значение без остановки выполнения кода

**Eloquent ORM**

Laravel содержит [ORM-библиотеку](https://ru.wikipedia.org/wiki/ORM) Eloquent, которая работает с базой данных. Eloquent использует связь Model со своей БД.

**find(id)** – возвращает элемент из БД соответствующий id

$user = User::find(1);

**create([])** –добавляет элементов в БД(в Model необходимо вставить protected $guarded=[] или protected $guarded=false;)

$user = User::create([

‘title’=>’some’,

…]);

**update([])** –изменяет существующий элементов в БД

$user = User::find(…)->update([

‘title’=>’update some’,

…]);

**delete()** –удаляет существующий элементов в БД(В Миграции(создание таблицы) указать $table->SoftDeletes()-данные останутся в таблице и их можно восстановить и в Model указать use SoftDeletes;)

$user = User::find(…)->delete();

**restore()** –восстанавливает удаленный элементов в БД(если он был удален при помощи SoftDeletes)

$user = User:: withTrashed() ->where('airline\_id', 1) ->restore()

**firstOrCreate([проверка], [новый элемент])** –возвращает элемент БД соответствующий условиям(например значение определенного атрибута), а если такого нет то создаст *Новый* и возвращает его. Можно использовать, чтобы избежать дублирования

**updateOrCreate([проверка], [обновление элемента или новый элемент])** – тоже что и firstOrCreate(), только если находится элемент соответствующий условию в него вносятся изменения и возвращает измененный элемент

**Методы Laravel**

**all()** –возвращает массив из Коллекция(collect). Позволяет получить все элементы БД

$user = User::all();

**where(‘ключ’, ‘значение’)** –фильтрует Коллекция(collect) по “ключ” / “значение”

$posts=User::where('is\_published', 1)->get();

-or-

$posts=User:: all()->where('is\_published', 1);

**first()** –возвращает первый элемент Коллекция(collect). Позволяет получить первый элемент БД

$user = User::first();

**Создание Controller**

**-**в командной строке php artisan make:controller [Name(CamelCase)Conroller]

**Создание Migration(DataBase) c Model**

**-**в командной строке php artisan make:model [Name] -m

-подключение к базе: файл .env заменить DB\_DATABASE на имя вашей базы

**Накатить(выполнить) Migration**

**-**в командной строке php artisan migrate

**Обновить Migration**

**-**в командной строке php artisan migrate:fresh