

Les commandes est les methods : laravel

create projet => composer create-project laravel/laravel example-app

lancer projet => php artisan serve

create model => php artisan make:model Test

create controller => php artisan make:controller TestController -r

create migration => php artisan make:migration create_test_table

Pour ajouter la table

```
return new class extends Migration{
    public function up(){
        Schema::create('test',function(Blueprint $table){
            $table->id();
            ... autre column
            $table->timestamps();
        })
    }
}
```

Pour supprimer la table

```
public function down(){
    Schema::dropIfExists('test');
}
}
```

ajouter column a table => php artisan make:migration add_column_to_test_table --table=test

```
public function up(){
    Schema::table('test',function(Blueprint $table){
        $table->string('nom_column');
    })
}
```

ajouter foreign key a table => php artisan make:migration add_column_id_to_test_table --table=test

```
Public function up(){
    Schema::table('test',function(Blueprint $table){
        $table->unsignedBigInteger('column_id');
        $table->foreign('column_id')->references('id')->on('nom_tabel')
    })
}
```

supprimer column a table => php artisan make:migration remove_column_from_test_table --table=test

```
Public function up(){
    Schema::table('test',function(Blueprint $table){
        $table->dropColumn('column');
    })
}
```

create model avec migration => php artisan make:model Test -m

create model avec migration et controller => php artisan make:model Test -mcr

create seeder => php artisan make:seeder TestSeeder

<pre> Class TestSeeder extends Seeder Public function run(){ DB::table('test')->insert(['nom'=>'moussan', ... autre data]) } </pre>	<p>Avec factory</p> <pre> Class TestSeeder extends Seeder Public function run(){ Test::factory(n fois)->create() } </pre> <p>n fois = 10 or 20 ...</p>
---------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------

create factory => php artisan make:factory TestFactory --model=Test

<pre> Class TestFactory extends Factory{ Public function definition(){ return['nom'=>fake()->name(), ... autre column] } } </pre>

Run seeder => php artisan db:seed --class=TestSeeder | php artisan db:seed

Empty database and run seeder => php artisan migrate:fresh --seed

create request => php artisan make:request TestRequest

create middleware => php artisan make:middleware TestMiddleware

Method handel	Method terminate
----------------------	-------------------------

<pre>Public function handle(Request \$request , Closure \$next){ Return \$next(request) }</pre>	<pre>Public function handle(Request \$request , Response \$response){ }</pre>
-----------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------

Routes =>

<p>Pour lister la vue index</p> <pre>Route::get('/index',[nom_controller,'index'])->name('index')</pre> <p>Pour la vue show</p> <pre>Route::get('/show/{test}',[nom_controller,'show'])->name('show')</pre> <p>Pour ajouter la vue create</p> <pre>Route::get('/create',[nom_controller,'create'])->name('create')</pre> <p>Pour modifier la vue edite</p> <pre>Route::get('/edit/{test}',[nom_controller,'edit'])->name('edit')</pre> <p>Pour souvgarder</p> <pre>Route::post('/store',[nom_controller,'store'])->name('store')</pre> <p>Pour update</p> <pre>Route::update('/update/{test}',[nom_controller,'update'])->name('update')</pre> <p>Pour delete</p> <pre>Route::delete('/destroy/{test}',[nom_controller,'destroy'])->name('destroy')</pre> <p>Pour voire tout ca dans une route</p> <pre>Route::resources('/test,nom_controller)</pre>	
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--

Relationships => BelongsTo | HasMany

<pre>class Habitan extends Model { Public function villes() :BelongsTo { Return \$this->BelongTo(Ville::class); }</pre>	<pre>class Ville extends Model { Public function habitants() :HasMany { Return \$this->HasMany(Habitan::class); }</pre>
--------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------

<pre> } } } </pre>	<pre> } } } </pre>
--------------------	--------------------

Importes namespace =>

App\Http\Controllers | App\Http\Middleware

App\Models

database\migrations | factories : database\factories | seeders : database\seeders

resources/views

routes=>web | console | api | channels

store() and update() photo =>

Dans le form create on ajouter attribute enctype="multipart/form-data"	Dans le form edite on ajouter attribute enctype="multipart/form-data"
Store() <pre> \$newtest = \$request->file('photo')->store('dossier','public') </pre>	Update() <pre> if(\$request->hasFile('photo')) { \$newtest = \$request->file('photo')->store('dossier','public') } </pre>

Laravel Breeze=>

1 : composer require laravel/breeze | 2 : php artisan breeze:install

3 : php artisan migrate | 4 : npm install | 5 : npm run dev

CRUD =>

<pre>public function index() { \$villes = Ville::all() return view('villes.index', compact('villes')) }</pre>	
<pre>public function create() { return view('villes.create') }</pre>	<pre>public function store(Request \$request) { \$newVille = new Ville(); \$newVille->nom = \$request->nom \$newVille->save() return redirect()->route('villes.index') }</pre>
<pre>public function show(Ville \$ville) { return view('villes.show', compact('ville')) }</pre>	
<pre>public function edit(Ville \$ville) { return view('villes.edit', compact('ville')) }</pre>	<pre>public function update(Request \$request, Ville \$ville) { \$ville->nom = \$request->nom \$ville->update() return redirect()->route('villes.index') }</pre>
<pre>public function destroy(Ville \$ville) { \$ville->delete return redirect()->route('villes.index') }</pre>	

