

### Project command

### update data tables

```
Update table add col php artisan make:migration add_comment_to_products_table $table->text('comment');
```

### SEED

```
Use factory on seed Product::factory(10)-->create();
run seed php artisan db:seed
run seed and migrate --seed php artisan migrate --seed
```

### ELOQUENT ORM

```
New Element $flight = new Flight;
Find Element $flight = Flight::find(1);
Update Element $flight->name = 'New Flight Name';
Save Element $flight->save();
Create inline $user = User::create(['first_name' => 'Taylor', 'last_name' => 'Otwell']);
Update All Flight::where('active', 1)->update(['delayed' => 1]);
Delete $current_user->delete();
Delete by id User::destroy(1);
Delete all $deletedRows = Flight::where('active', 0)->delete();
Get All $items = Item::all();
Find one by primary key $flight = Flight::find(1);
display 404 if not found $model = Flight::findOrFail(1);
```

### ELOQUENT ORM (cont)

```
Get last entry $items = Item::latest()->get();
Chain $flights = App\Flight::where('active', 1)->orderBy('name', 'desc')->take(10)->get();
Where Todo::where('id', $id)->findOrFail();
Like Todos::where('name', 'like', '%'. $my . '%')->get();
Or where Todos::where('name', 'mike')->orWhere('title', '=', 'Admin')->get();
Count $count = Flight::where('active', 1)->count();
Sum $sum = Flight::where('active', 1)->sum('price');
Contain if ($project->$users->contains('mike'))
```

### Create a table

```
Create data table php artisan make:migration create_products_table
Foreign Key relation $table->foreignId('user_id')->constrained('users')->onDelete('cascade');
Default value $table->boolean('isActive')->default(true);
Not required $table->text('description')->nullable();
Unique constraint $table->string('modelNo')->unique();
```

### Factory

```
database/factories/ProductFactory.php
public function definition() {
    return [ 'name' => $this->faker->text(20), 'price' => $this->faker->numberBetween(10, 10000), ];
}
```

### Routes

```
Basic route closure Route::get('/greeting', function () { return 'Hello World'; });
Route direct view shortcut Route::view('/welcome', 'welcome');
Route to controller class Route::get('/user', 'UserController@index');
Route only for specific HTTP verbs Route::match(['get', 'post'], '/', function () { // });
Route for all verbs Route::any('/', function () { // });
Route Redirect Route::redirect('/clients', '/customers');
Route Parameters Route::get('/user/{id}', function ($id) { return 'User '.$id; });
Optional Parameter Route::get('/user/{name?}', function ($name = 'John') { return $name; });
```

### Routes (cont)

Named Route  
Route::get('/user/profile', [UserProfileController::class, 'show'] )->name('profile');

Resource Route  
Route::resource('photos', PhotoController::class);

Resource Short  
Route::resource('photos', 'PhotoController');

URL with route name  
\$url = route('profile', ['id' => 1]);

Generating Redirects...  
return redirect()->route('profile');

Route model binding  
Route::get('/users/{user}', function (User \$user) { return \$user->email; });

### Controller

Set validation rules  
protected \$rules = [ 'title' => 'required|unique:posts|max:255', 'name' => 'required|min:6'];

Validate  
\$validatedData = \$request->validate(\$rules)

Show 404 error page  
abort(404, 'Sorry, Post not found')

### Artisan common

Database migration  
php artisan migrate

Data seed  
php artisan db:seed --class=NomeClasse

Create from model  
php artisan make:model Product

### Artisan common (cont)

Option for model  
-m (migration), -c (controller), -r (resource controllers), -f (factory), -s (seed)

Create a controller  
php artisan make:controller ProductsController

Update table migration  
php artisan make:migration add\_date\_to\_blogposts\_table

Rollback latest migration  
php artisan migrate:rollback

Rollback all migrations  
php artisan migrate:reset

Rollback all and re-migrate  
php artisan migrate:refresh

Rollback all, re-migrate and seed  
php artisan migrate:refresh --seed

### HELPERS

Display variable content and kill execution  
dd(\$products)

Create a Laravel collection from array.  
\$collection = collect(\$array);

Sort by description ascending  
\$ordered\_collection = \$collection->orderBy('description');

Reset numbering value  
\$ordered\_collection = \$ordered\_collection->values()->all();

### Install a project from github

```
git clone {project http address} projectName
cd projectName
```

```
composer install
```

```
cp .env.example .env
```

```
php artisan key:generate
```

```
php artisan migrate
```

```
npm install
```

### Hot reload

```
npm install
```

Open webpack.mix.js

```
mix.browserSync('127.0.0.1:8000')
```

```
npm install browser-sync browser-sync-webpack-plugin
```

```
php artisan serve
```

```
npm run watch
```



By **FrancoSvalvast**

[cheatography.com/francosvalvast/](https://cheatography.com/francosvalvast/)

Not published yet.

Last updated 12th November, 2022.

Page 2 of 2.

Sponsored by **ApolloPad.com**

Everyone has a novel in them. Finish Yours!

<https://apollopod.com>