Referencia: https://blog.mxcursos.com/laravel-5-2-como-instalar-o-laravel/

Ir na pasta do htdocs e dar o seguinte comando pelo cmd

composer create-project --prefer-dist laravel/laravel blog

alterar permissões dos diretórios boostrap/cache e storage

para acessar o projeto que foi criado deve ir na pasta onde esta o projeto e dar o seguinte comando:

php artisan serve

php artisan app:name estoque – para trocar o nome da pasta do namespace

criar novo projeto laravel new estoque

Para escolher a versão que sera usada ao criar o projeto - composer create-project laravel/laravel estoque "5.0."

php artisan make:model

Para colocar o nome real da tabel no banco de dados e desativar campos que são criados automaticamente pelo eloquent

protected $table = 'produtos';

public $timestamps = false;

para deixar salvar apenas esses campos por requisição

protected fillable = array('nome', 'descricao', 'valor', 'quantidade');

ao tentar salvar dados no banco, não deixa alterar o valor do campo id

protected $guarded = ['id'];

para validar campos de um formulário com o laravel deve dar o seguinte comando na pasta dp projeto

php artisan make:request ProdutosRequest

depois abrir o arquivo e alterar o primeiro método para true, o segundo sera onde vai ter as regras de validação

como abaixo:

return [

'nome' => 'required|max:100',

'descricao' => 'required|max:255',

'valor' => 'required|numeric'

];

Para alterar as mensagens de validação deve criar o método abaixo nesse mesmo arquivo

public function messages(){

return [

'required' => 'The :attribute field can not be empty.'

'nome.required' => 'The :attribute field can not be empty.',

];

}

Para essa classe ser utilizada passar como parâmetro no método que sera utilizado para inserçõa dos dados, Como abaixo

use Request;

use estoque\Http\Requests\ProdutosRequest;

public function adiciona(ProdutosRequest $request){

Produto::create($request->all());

return redirect()

->action('ProdutoController@lista')

->withInput(Request::only('nome'));

}

Para exibir os erros de validação na view é utilizado o código abaixo

@foreach($errors->all() as $error)

{{ $error }}

@endforeach

Caso haja necessidade as mensagens de erro podem ser aplicadas ao projeto todo, para isso deve ser criar as regras em custom no arquivo validation.php do projeto

'custom' => [

'attribute-name' => [

'rule-name' => 'custom-message',

],

],

Para não perder os valores dos campos ao não ser validado o formulário no input colocar o seguinte:

<input name="nome" value="{{ old('nome') }}" />

Para verificar se deu erro a validaçõa

if ($validator->fails())

{

return redirect()->action('ProdutoController@novo');

}