Contenido

[Connection of database 3](#_Toc481069287)

[Crud with model eloquent 3](#_Toc481069288)

[Relations of model 4](#_Toc481069289)

[One to many 4](#_Toc481069290)

[Many to many 6](#_Toc481069291)

[Controller 7](#_Toc481069292)

[public function index() 9](#_Toc481069293)

[public function create() 9](#_Toc481069294)

[public function store(Request $request) 10](#_Toc481069295)

[public function show($id) 11](#_Toc481069296)

[public function edit($id) 11](#_Toc481069297)

[public function update(Request $request, $id) 12](#_Toc481069298)

[public function destroy($id) 12](#_Toc481069299)

[Views 13](#_Toc481069300)

[Layout 13](#_Toc481069301)

[Index view 14](#_Toc481069302)

[Create view 16](#_Toc481069303)

[Edit view 18](#_Toc481069304)

[Delete view 20](#_Toc481069305)

[Validation 21](#_Toc481069306)

[Validation in controller 22](#_Toc481069307)

[Validation of multiple select 23](#_Toc481069308)

[Validation of many input arrays 23](#_Toc481069309)

[Custom error messsages 23](#_Toc481069310)

[Pagination 24](#_Toc481069311)

[Security 25](#_Toc481069312)

[Tables 25](#_Toc481069313)

[Models 25](#_Toc481069314)

[User model 26](#_Toc481069315)

[Rol Model 27](#_Toc481069316)

[LogueoController 28](#_Toc481069317)

[Route 30](#_Toc481069318)

[Usuario Controller 30](#_Toc481069319)

[Views 35](#_Toc481069320)

[Login view 35](#_Toc481069321)

[Usuario 36](#_Toc481069322)

[The most important thing 40](#_Toc481069323)

[Crud with query builder 41](#_Toc481069324)

[Create the controller 41](#_Toc481069325)

[Creating the views 42](#_Toc481069326)

[Layout 42](#_Toc481069327)

[Index view 43](#_Toc481069328)

[Create view 45](#_Toc481069329)

[Edit view 47](#_Toc481069330)

[Creating index in controller 50](#_Toc481069331)

[Creating insert in controller 50](#_Toc481069332)

[Creating store in controller 50](#_Toc481069333)

[Creating edit in controller 51](#_Toc481069334)

[Creating update in controller 52](#_Toc481069335)

[Creating destroy in controller 53](#_Toc481069336)

[Script 53](#_Toc481069337)

[Create a rute of specific method of controller 55](#_Toc481069338)

[Create flash message 55](#_Toc481069339)

[Old values in form 55](#_Toc481069340)

# Connection of database

Open .env and configure the parameters necessary for the connection of database

# Crud with model eloquent

Configure the file .ev in laravel to connect to the database

Create the models using

php artisan make:model Producto

The models are in app/

The second is you need to put the name of table and quit timestamp

/\*\*

\* The table associated with the model.

\*

\* @var string

\*/

protected $table = 'producto';

protected $primaryKey = 'idproducto';

/\*\*

\* Indicates if the model should be timestamped.

\*

\* @var bool

\*/

public $timestamps = false;

## Relations of model

### One to many

One Tienda has many Productos

class Tienda extends Model

{

/\*\*

\* The table associated with the model.

\*

\* @var string

\*/

protected $table = 'tienda';

protected $primaryKey = 'idtienda';

/\*\*

\* Indicates if the model should be timestamped.

\*

\* @var bool

\*/

public $timestamps = false;

public function productos()

{

return $this->hasMany('App\Producto');

}

}

//One product belongs to one tienda

class Producto extends Model

{

/\*\*

\* The table associated with the model.

\*

\* @var string

\*/

protected $table = 'producto';

protected $primaryKey = 'idproducto';

/\*\*

\* Indicates if the model should be timestamped.

\*

\* @var bool

\*/

public $timestamps = false;

public function tienda()

{

return $this->belongsTo('App\Tienda','idtienda');

}

}

### Many to many

One product has many categories and one categories belons to many products

class Producto extends Model

{

/\*\*

\* The table associated with the model.

\*

\* @var string

\*/

protected $table = 'producto';

protected $primaryKey = 'idproducto';

/\*\*

\* Indicates if the model should be timestamped.

\*

\* @var bool

\*/

public $timestamps = false;

public function tienda()

{

return $this->belongsTo('App\Tienda','idtienda');

}

//producto\_categoria is the intermedium table, idproducto and idcategoria are primary and foreign key (Producto and Categoria)

public function categorias()

{

return $this->belongsToMany('App\Categoria','producto\_categoria','idproducto','idcategoria');

}

}

### One to One

It’s the same about was mentioned before

public function config()

{

return $this->hasOne('App\TtrconfigField','ttrfield\_id');

}

public function field()

{

return $this->belongsTo('App\Ttrfield','ttrfield\_id');

}

### One important thing.

In controller you have

public function edit($id)

{

$formulario = Ttrform::where('idttrform','=',$id)->first();

$cargos = Cargo::all();

return view('formulario.edit',array('formulario'=>$formulario,'cargos'=>$cargos));

}

In view

@foreach ($formulario->fields as $ttrfield)

after

@foreach ($ttrfield->values as $valor)

In the models is of formulario

public function fields()

{

return $this->hasMany('App\Ttrfield','ttrform\_id');

}

In fields is

public function values()

{

return $this->hasMany('App\Ttrvalue','idttrfieldsf');

}

public function formulario()

{

return $this->belongsTo('App\Ttrform','ttrform\_id');

}

And in values is

public function field()

{

return $this->belongsTo('App\Ttrfield','idttrfieldsf');

}

#### When you have relations you must put the foreign . For example.

One form can have many fields

public function fields()

{

return $this->hasMany('App\Ttrfield','ttrform\_id');

}

Look the foreign key is 'ttrform\_id'. It’s the second parameter

and one field belongs to formulario

public function formulario()

{

return $this->belongsTo('App\Ttrform','ttrform\_id');

}

Look that 'ttrform\_id' is the foreign key of fields

One filed can have many values

public function values()

{

return $this->hasMany('App\Ttrvalue','idttrfieldsf');

}

One value belongs to a field

public function field()

{

return $this->belongsTo('App\Ttrfield','idttrfieldsf');

}

So

@foreach ($formulario->fields as $ttrfield)

@foreach ($ttrfield->values as $valor)

This is the way to call one table that is related with another table.

When you need to call this put

This is other thing

$ttrfield->config->required

Without pharentesis

## Controller

First execute this line in cmd

php artisan make:controller ProductoController --resource

This will create the controller

The second is put this routes/web.php

Route::resource('productos', 'ProductoController');

This will create these default routes as you can see

|  |  |  |  |
| --- | --- | --- | --- |
| GET | /photos | index | photos.index |
| GET | /photos/create | create | photos.create |
| POST | /photos | store | photos.store |
| GET | /photos/{photo} | show | photos.show |
| GET | /photos/{photo}/edit | edit | photos.edit |
| PUT/PATCH | /photos/{photo} | update | photos.update |
| DELETE | /photos/{photo} | destroy | photos.destroy |

The next is the ProductoController

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

use App\Producto as Producto;

use App\Tienda as Tienda;

use App\Categoria as Categoria;

class ProductoController extends Controller

{

/\*\*

\* Display a listing of the resource.

\*

\* @return \Illuminate\Http\Response

\*/

### public function index()

{

//

$productos = Producto::all();

return view('producto.index',['productos'=>$productos]);

}

/\*\*

\* Show the form for creating a new resource.

\*

\* @return \Illuminate\Http\Response

\*/

### public function create()

{

//

$tiendas = Tienda::all();

$categorias = Categoria::all();

return view('producto.create',['tiendas'=>$tiendas,'categorias'=>$categorias]);

}

/\*\*

\* Store a newly created resource in storage.

\*

\* @param \Illuminate\Http\Request $request

\* @return \Illuminate\Http\Response

\*/

### public function store(Request $request)

{

$producto = new Producto;

$producto->nombre = $request->nombre;

$producto->precio = $request->precio;

$producto->idtienda = $request->tienda;

$producto->save();

$categorias = $request->categoria;

$producto->categorias()->attach($categorias);

//despues de guardar retorne a index

$productos = Producto::all();

return redirect('productos');

}

/\*\*

\* Display the specified resource.

\*

\* @param int $id

\* @return \Illuminate\Http\Response

\*/

### public function show($id)

{

$producto = Producto::where('idproducto','=',$id)->first();

return view('producto.delete',['producto'=>$producto]);

}

/\*\*

\* Show the form for editing the specified resource.

\*

\* @param int $id

\* @return \Illuminate\Http\Response

\*/

### public function edit($id)

{

$producto = Producto::where('idproducto','=',$id)->first();

$tiendas = Tienda::all();

$categorias = Categoria::all();

return view('producto.edit',['producto'=>$producto,'tiendas'=>$tiendas,'categorias'=>$categorias]);

}

/\*\*

\* Update the specified resource in storage.

\*

\* @param \Illuminate\Http\Request $request

\* @param int $id

\* @return \Illuminate\Http\Response

\*/

### public function update(Request $request, $id)

{

$producto = Producto::where('idproducto','=',$id)->first();

$producto->nombre = $request->nombre;

$producto->precio = $request->precio;

$producto->idtienda = $request->tienda;

$producto->save();

$categorias = $request->categoria;

$producto->categorias()->detach($producto->categorias);

$producto->categorias()->attach($categorias);

//despues de guardar retorne a index

return redirect('productos');

}

/\*\*

\* Remove the specified resource from storage.

\*

\* @param int $id

\* @return \Illuminate\Http\Response

\*/

### public function destroy($id)

{

$producto = Producto::where('idproducto','=',$id)->first();

$producto->categorias()->detach();

$producto->delete();

//despues de guardar retorne a index

$productos = Producto::all();

return redirect('productos');

}

}

## Views

Create a folder with the name product and create these files

Index.blade.php, create.blade.php, update.blade.php

### Layout

Create a folder with the name layouts and within views and create template.blade.php into layouts

<html>

<head>

<title>App Name - @yield('title')</title>

</head>

<body>

@section('sidebar')

This is the master sidebar.

@show

<div class="container">

@yield('content')

</div>

</body>

</html>

#### Include scripts in template

Put in your layout

@yield('footer\_scripts')

And in the template put

@section('footer\_scripts')

<script src="{{ asset('js/form.js?v=1') }}"></script>

<script src="{{ asset('js/formulario\_ajax.js') }}"></script>

@stop

### Index view

@extends('layouts.template1')

@section('title', 'Page Title')

@section('sidebar')

@parent

<p>This is appended to the master sidebar.</p>

@endsection

@section('content')

<a href="{{ url('productos/create') }}">Crear producto</a>

<table>

<th>Nombre</th>

<th>Precio</th>

<th>Tienda</th>

<th>Categorias</th>

<th>Acciones</th>

@foreach ($productos as $producto)

<tr>

<td>{{ $producto->nombre }}</td>

<td>{{ $producto->precio }}</td>

<td>{{ $producto->tienda->nombre }}</td>

<td>

@foreach ($producto->categorias as $categoria)

{{ $categoria->nombre }},

@endforeach

</td>

<td>

<a href="{{ url('productos/'.$producto->idproducto.'/edit') }}">Editar</a>

<a href="{{ url('productos/'.$producto->idproducto) }}">Eliminar</a>

</td>

</tr>

@endforeach

</table>

@endsection

### Create view

@extends('layouts.template1')

@section('title', 'Page Title')

@section('sidebar')

@parent

<p>This is appended to the master sidebar.</p>

@endsection

@section('content')

<form action="{{ url('productos') }}" method="POST">

<div>

<label>Nombre</label>

<input type="text" name="nombre" value="{{ old('nombre') }}">

</div>

<div>

<label>Precio</label>

<input type="text" name="precio" value="{{ old('precio') }}">

</div>

<div>

<label>Tienda</label>

<select name="tienda">

<option value="">Seleccione la tienda</option>

@foreach ($tiendas as $tienda)

<option value="{{ $tienda->idtienda }}">{{ $tienda->nombre }}</option>

@endforeach

</select>

</div>

<div>

<label>Categorias</label>

<select name="categoria[]" multiple>

<option value="">Seleccione la tienda</option>

@foreach ($categorias as $categorias)

<option value="{{ $categorias->idcategoria }}">{{ $categorias->nombre }}</option>

@endforeach

</select>

</div>

<div>

<input type="submit" name="">

</div>

<input type="hidden" name="\_token" value="{{ csrf\_token() }}">

</form>

@endsection

### Edit view

@extends('layouts.template1')

@section('title', 'Page Title')

@section('sidebar')

@parent

<p>This is appended to the master sidebar.</p>

@endsection

@section('content')

<form action="{{ url('/productos/'.$producto->idproducto) }}" method="POST">

<div>

<label>Nombre</label>

<input type="text" name="nombre" value="{{ $producto->nombre }}">

</div>

<div>

<label>Precio</label>

<input type="text" name="precio" value="{{ $producto->precio }}">

</div>

<div>

<label>Tienda</label>

<select name="tienda">

<option value="">Seleccione la tienda</option>

@foreach ($tiendas as $tienda)

<option @if($tienda->idtienda == $producto->idtienda) selected @endif value = "{{ $tienda->idtienda }}">{{ $tienda->nombre }}</option>

@endforeach

</select>

</div>

<div>

<label>Categorias</label>

<select name="categoria[]" multiple>

<option value="">Seleccione la tienda</option>

@foreach ($categorias as $categoria)

<option value="{{ $categoria->idcategoria }}"

@foreach ($producto->categorias as $categoria\_producto)

@if($categoria\_producto->idcategoria == $categoria->idcategoria) selected @endif

@endforeach

>{{ $categoria->nombre }}</option>

@endforeach

</select>

</div>

<div>

<input type="submit" name="">

</div>

<input type="hidden" name="\_token" value="{{ csrf\_token() }}">

<input name="\_method" type="hidden" value="PATCH">

</form>

@endsection

### Delete view

@extends('layouts.template1')

@section('title', 'Page Title')

@section('sidebar')

@parent

<p>This is appended to the master sidebar.</p>

@endsection

@section('content')

<form action="{{ url('/productos/'.$producto->idproducto) }}" method="POST">

<div>

<h1>Esta seguro de borrar este producto?</h1>

</div>

<div>

<h2>{{ $producto->nombre }}</h2>

</div>

<div>

<input type="submit" name="" value="borrar">

</div>

<input type="hidden" name="\_token" value="{{ csrf\_token() }}">

<input name="\_method" type="hidden" value="DELETE">

</form>

@endsection

# Validation

public function store(Request $request)

{

$this->validate($request, [

'title' => 'required|unique:posts|max:255',

'body' => 'required',

]);

// The blog post is valid, store in database...

}

## Validation in controller

$validator = Validator::make($request->all(), [

'nombre\_formulario' => 'required',

'formulario\_activo' => 'required',

'cargos[]' => 'required',

]);

if ($validator->fails())

{

$errors = $validator->errors();

foreach ($errors->all() as $message) {

echo "<li>".$message."</li>";

}

return;

}

In template put

@if (count($errors) > 0)

<div class="alert alert-danger">

<ul>

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

<li>{{ $error }}</li>

@endforeach

</ul>

</div>

@endif

## Validation of multiple select

'cargos' => 'required|array|min:1',

## Validation of many input arrays

This will validate all the names that contains name=nombre\_campo[]

'nombre\_campo.\*' => 'required',

## Custom error messsages

$messages = [

'required' => 'El :attribute es requerido.',

];

$validator = Validator::make($request->all(), [

'nombre\_formulario' => 'required',

'formulario\_activo' => 'required',

'cargos' => 'required|array|min:1',

'nombre\_campo.\*' => 'required',

'tipo\_campo.\*' => 'required',

'campo\_activo.\*' => 'required',

'campo\_requerido.\*' => 'required',

],$messages);

# Pagination

$users = App\User::paginate(15);

In the view

<div class="container">

@foreach ($users as $user)

{{ $user->name }}

@endforeach

</div>

{{ $users->links() }}

# Security

The tables that has been used are

Important . It’s required the field password and remember\_token in the user table in order to use the security laravel

One user can has many roles.

## Tables

|  |  |
| --- | --- |
| usuarios | CREATE TABLE `usuarios` (  `idusuario` int(11) NOT NULL AUTO\_INCREMENT,  `nombre` varchar(100) NOT NULL,  `email` varchar(255) NOT NULL,  `telefono` varchar(100) DEFAULT NULL,  `password` varchar(255) NOT NULL,  `remember\_token` varchar(255) DEFAULT NULL,  PRIMARY KEY (`idusuario`) ) |

CREATE TABLE `roles` (  
 `idrol` int(11) NOT NULL AUTO\_INCREMENT,  
 `nombre` varchar(100) NOT NULL,  
 PRIMARY KEY (`idrol`)  
)

CREATE TABLE `usuarios\_roles` (  
 `idusuario` int(11) NOT NULL,  
 `idrol` int(11) NOT NULL,  
 PRIMARY KEY (`idusuario`,`idrol`),  
 KEY `idrol` (`idrol`),  
 CONSTRAINT `usuarios\_roles\_ibfk\_1` FOREIGN KEY (`idusuario`) REFERENCES `usuarios` (`idusuario`),  
 CONSTRAINT `usuarios\_roles\_ibfk\_2` FOREIGN KEY (`idrol`) REFERENCES `roles` (`idrol`)  
)

## Models

### User model

<?php

namespace App;

use Illuminate\Notifications\Notifiable;

use Illuminate\Foundation\Auth\User as Authenticatable;

class User extends Authenticatable

{

use Notifiable;

/\*\*

\* The attributes that are mass assignable.

\*

\* @var array

\*/

protected $fillable = [

'name', 'email', 'password',

];

/\*\*

\* The attributes that should be hidden for arrays.

\*

\* @var array

\*/

protected $hidden = [

'password', 'remember\_token',

];

protected $table = 'usuarios';

protected $primaryKey = 'idusuario';

public $timestamps = false;

public function roles()

{

return $this->belongsToMany('App\Rol','usuarios\_roles','idusuario','idrol');

}

}

### Rol Model

<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class Rol extends Model

{

/\*\*

\* The table associated with the model.

\*

\* @var string

\*/

protected $table = 'roles';

protected $primaryKey = 'idrol';

/\*\*

\* Indicates if the model should be timestamped.

\*

\* @var bool

\*/

public $timestamps = false;

}

## LogueoController

<?php

namespace App\Http\Controllers;

use Illuminate\Support\Facades\Auth;

use Illuminate\Http\Request;

class LogueoController extends Controller

{

/\*\*

\* Handle an authentication attempt.

\*

\* @return Response

\*/

public function authenticate(Request $request)

{

if (Auth::attempt(['email' => $request->usuario, 'password' => $request->clave ],$request->recuerdame)) {

// Authentication passed...

return redirect()->intended('usuarios');

}

else

{

echo "Credenciales incorrectas";

}

}

public function index()

{

return view('login.index');

}

public function cerrar\_sesion()

{

Auth::logout();

return view('login.index');

}

}

### Route

In routes/web.php put

Route::post('autenticar', 'LogueoController@authenticate');

Route::get('cerrar\_sesion', 'LogueoController@cerrar\_sesion');

Route::get('ingreso', 'LogueoController@index');

### Usuario Controller

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

use App\User;

use App\Rol;

use Illuminate\Support\Facades\Auth;

class UsuarioController extends Controller

{

/\*\*

\* Display a listing of the resource.

\*

\* @return \Illuminate\Http\Response

\*/

public function index()

{

if((!Auth::viaRemember() && !Auth::check()) || (!Auth::User()->roles()->where('nombre' , 'admin')->exists() &&

!Auth::User()->roles()->where('nombre' , 'ver')->exists()))

{

return view('invalido');

}

$usuarios = User::all();

return view('usuario.index',['usuarios'=>$usuarios]);

}

/\*\*

\* Show the form for creating a new resource.

\*

\* @return \Illuminate\Http\Response

\*/

public function create()

{

if(!Auth::check() || (!Auth::User()->roles()->where('nombre' , 'admin')->exists() &&

!Auth::User()->roles()->where('nombre' , 'insertar')->exists()))

{

return view('invalido');

}

$roles = Rol::all();

return view('usuario.create',['roles'=>$roles]);

}

/\*\*

\* Store a newly created resource in storage.

\*

\* @param \Illuminate\Http\Request $request

\* @return \Illuminate\Http\Response

\*/

public function store(Request $request)

{

if(!Auth::check() || Auth::User()->roles()->whereNotIn('nombre' , ['admin','insertar']))

{

return view('invalido');

}

$usuario = new User;

$usuario->nombre = $request->nombre;

$usuario->email = $request->email;

$usuario->telefono = $request->telefono;

$usuario->password = bcrypt($request->clave);

$usuario->save();

$roles = $request->rol;

$usuario->roles()->attach($roles);

return redirect('usuarios');

}

/\*\*

\* Display the specified resource.

\*

\* @param int $id

\* @return \Illuminate\Http\Response

\*/

public function show($id)

{

//

}

/\*\*

\* Show the form for editing the specified resource.

\*

\* @param int $id

\* @return \Illuminate\Http\Response

\*/

public function edit($id)

{

//

}

/\*\*

\* Update the specified resource in storage.

\*

\* @param \Illuminate\Http\Request $request

\* @param int $id

\* @return \Illuminate\Http\Response

\*/

public function update(Request $request, $id)

{

//

}

/\*\*

\* Remove the specified resource from storage.

\*

\* @param int $id

\* @return \Illuminate\Http\Response

\*/

public function destroy($id)

{

//

}

}

## Views

### Login view

The login is in the folder Login

@extends('layouts.template1')

@section('content')

<form action="{{ url('autenticar') }}" method="post">

<div>

<label>Usuario</label>

<input type="text" name="usuario">

</div>

<div>

<label>Clave</label>

<input type="password" name="clave">

</div>

<div>

<input type="checkbox" name="recuerdame">Recuerdame

</div>

<div>

<input type="submit" name="">

</div>

<input type="hidden" name="\_token" value="{{ csrf\_token() }}">

</form>

@endsection

### Usuario

#### Index view

@extends('layouts.template1')

@section('content')

@if((Auth::User()->roles()->where('nombre' , 'admin')->exists() ||

Auth::User()->roles()->where('nombre' , 'insertar')->exists()))

<a href="{{ url('usuarios/create') }}">Crear Usuario</a>

@endif

<table>

<th>Nombre</th>

<th>Email</th>

<th>Telefono</th>

<th>Roles</th>

<th>Acciones</th>

@foreach ($usuarios as $usuario)

<tr>

<td>{{ $usuario->nombre }}</td>

<td>{{ $usuario->email }}</td>

<td>{{ $usuario->telefono }}</td>

<td>

@foreach ($usuario->roles as $rol)

{{ $rol->nombre }},

@endforeach

</td>

<td>

<a href="{{ url('usuarios/'.$usuario->idusuario.'/edit') }}">Editar</a>

<a href="{{ url('usuarios/'.$usuario->idusuario) }}">Eliminar</a>

</td>

</tr>

@endforeach

</table>

@endsection

#### Create view

@extends('layouts.template1')

@section('content')

<form action="{{ url('usuarios') }}" method="POST">

<div>

<label>Nombre</label>

<input type="text" name="nombre" value="{{ old('nombre') }}">

</div>

<div>

<label>Email</label>

<input type="text" name="email" value="{{ old('email') }}">

</div>

<div>

<label>Telefono</label>

<input type="text" name="telefono" value="{{ old('telefono') }}">

</div>

<div>

<label>Clave</label>

<input type="password" name="clave" value="{{ old('clave') }}">

</div>

<div>

<label>Roles</label>

@foreach ($roles as $rol)

<input type="checkbox" name="rol[]" value="{{ $rol->idrol }}">{{ $rol->nombre }}

@endforeach

</div>

<div>

<input type="submit" name="">

</div>

<input type="hidden" name="\_token" value="{{ csrf\_token() }}">

</form>

@endsection

## The most important thing

To log in the user is necessary to have

public function authenticate(Request $request)

{

if (Auth::attempt(['email' => $request->usuario, 'password' => $request->clave ],$request->recuerdame)) {

// Authentication passed...

return redirect()->intended('usuarios');

}

else

{

echo "Credenciales incorrectas";

}

}

The log out is

Auth::logout();

$request->recuerdame is a checkbox just to remember the user credentials that are store in cookies.

The permissions are

if((!Auth::viaRemember() && !Auth::check()) || (!Auth::User()->roles()->where('nombre' , 'admin')->exists() &&

!Auth::User()->roles()->where('nombre' , 'ver')->exists()))

This part (!Auth::viaRemember() && !Auth::check()) means if the user does not have remember and is not auth

# Where with relations

$users = App\User::with(['posts' => function ($query) {

$query->where('title', 'like', '%first%');

}])->get();

Other example

$formulario = Ttrform::with(['fields' => function ($query) {

$query->where('active', '=', '1');

}])

->where('idttrform','=',$id)

->first();

# Crud with query builder

## Create the controller

First execute this line in cmd

php artisan make:controller ProductoController --resource

This will create the controller

The second is put this routes/web.php

Route::resource('productos', 'ProductoController');

This will create these default routes as you can see

|  |  |  |  |
| --- | --- | --- | --- |
| GET | /photos | index | photos.index |
| GET | /photos/create | create | photos.create |
| POST | /photos | store | photos.store |
| GET | /photos/{photo} | show | photos.show |
| GET | /photos/{photo}/edit | edit | photos.edit |
| PUT/PATCH | /photos/{photo} | update | photos.update |
| DELETE | /photos/{photo} | destroy | photos.destroy |

## Creating the views

Create a folder with the name product and create these files

Index.blade.php, create.blade.php, update.blade.php

### Layout

Create a folder with the name layouts and within views and create template.blade.php into layouts

<html>

<head>

<title>App Name - @yield('title')</title>

</head>

<body>

@section('sidebar')

This is the master sidebar.

@show

<div class="container">

@yield('content')

</div>

</body>

</html>

### Index view

@extends('layouts.template1')

@section('title', 'Page Title')

@section('sidebar')

@parent

<p>This is appended to the master sidebar.</p>

@endsection

@section('content')

<a href="{{ url('productos/create') }}">Crear producto</a>

<table>

<th>Nombre</th>

<th>Precio</th>

<th>Categoria</th>

<th>Acciones</th>

@foreach ($productos as $producto)

<tr>

<td>{{ $producto->nombre }}</td>

<td>{{ $producto->precio }}</td>

<td>{{ $producto->cnombre }}</td>

<td>

<a href="{{ url('productos/'.$producto->id.'/edit') }}">Editar</a>

<a style="cursor:pointer;" data-t="{{ url('productos/'.$producto->id) }}" data-token="{{ csrf\_token() }}" onclick="eliminar(this)">Eliminar</a>

</td>

</tr>

@endforeach

</table>

@endsection

### Create view

@extends('layouts.template1')

@section('title', 'Page Title')

@section('sidebar')

@parent

<p>This is appended to the master sidebar.</p>

@endsection

@section('content')

<form action="{{ url('productos') }}" method="POST">

<div>

<label>Nombre del producto</label>

<input type="text" name="nombre" required>

</div>

<div>

<label>precio</label>

<input type="text" name="precio" required>

</div>

<div>

<label>Categoria</label>

<select name="categoria" required>

<option value="">Seleccione.....</option>

@foreach ($categorias as $categoria)

<option value = "{{ $categoria->id }}">{{ $categoria->nombre }}</option>

@endforeach

</select>

</div>

<div>

<input type="submit" name="Crear">

</div>

<input type="hidden" name="\_token" value="{{ csrf\_token() }}">

</form>

@endsection

@if (count($errors) > 0)

<div class="alert alert-danger">

<ul>

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

<li>{{ $error }}</li>

@endforeach

</ul>

</div>

@endif

### Edit view

@extends('layouts.template1')

@section('title', 'Page Title')

@section('sidebar')

@parent

<p>This is appended to the master sidebar.</p>

@endsection

@section('content')

<form action="{{ url('/productos/'.$producto->id) }}" method="POST">

<div>

<label>Nombre del producto</label>

<input type="text" name="nombre" value="{{ $producto->nombre }}" required>

</div>

<div>

<label>precio</label>

<input type="text" name="precio" value="{{ $producto->precio }}" required>

</div>

<div>

<label>Categoria</label>

<select name="categoria" required>

<option value="">Seleccione.....</option>

@foreach ($categorias as $categoria)

<option @if($categoria->id == $producto->idcategoria) selected @endif value = "{{ $categoria->id }}">{{ $categoria->nombre }}</option>

@endforeach

</select>

</div>

<div>

<input type="submit" value="Actualizar">

</div>

<input type="hidden" name="\_token" value="{{ csrf\_token() }}">

<input name="\_method" type="hidden" value="PATCH">

</form>

@endsection

@if (count($errors) > 0)

<div class="alert alert-danger">

<ul>

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

<li>{{ $error }}</li>

@endforeach

</ul>

</div>

@endif

## Creating index in controller

Put the scripts that are required

use Illuminate\Http\Request;

use Illuminate\Support\Facades\DB;

In controller index put this

public function index()

{

//

$productos = DB::table('productos')

->join('categorias', 'productos.idcategoria', '=', 'categorias.id')

->select('productos.\*', 'categorias.nombre as cnombre')

->get();

return view('producto.index', ['productos' => $productos]);

}

## Creating insert in controller

public function create()

{

//

$categorias = DB::table('categorias')->get();

return view('producto.create', ['categorias' => $categorias]);

}

## Creating store in controller

public function store(Request $request)

{

//

$nombre = $request->nombre;

$precio = $request->precio;

$idcategoria = $request->categoria;

$this->validate($request, [

'nombre' => 'required',

'precio' => 'required',

'categoria' => 'required',

]);

// si no hay errores en validate se inserta el registro

DB::table('productos')->insert(

['nombre' => $nombre, 'precio' => $precio , 'idcategoria'=> $idcategoria]

);

$productos = DB::table('productos')

->join('categorias', 'productos.idcategoria', '=', 'categorias.id')

->select('productos.\*', 'categorias.nombre as cnombre')

->get();

return view('producto.index', ['productos' => $productos]);

}

## Creating edit in controller

public function edit($id)

{

//

$categorias = DB::table('categorias')->get();

$producto = DB::table('productos')->where('id', $id)->first();

return view('producto.edit', ['producto' => $producto, 'categorias' => $categorias]);

}

## Creating update in controller

public function update(Request $request, $id)

{

//

$nombre = $request->nombre;

$precio = $request->precio;

$idcategoria = $request->categoria;

$this->validate($request, [

'nombre' => 'required',

'precio' => 'required',

'categoria' => 'required',

]);

DB::table('productos')

->where('id', $id)

->update(['nombre' => $nombre,'precio' =>$precio,'idcategoria'=>$idcategoria]);

return view('producto.edit', ['producto' => $producto, 'categorias' => $categorias]);

}

## Creating destroy in controller

public function destroy($id)

{

//

DB::table('productos')->where('id', '=', $id)->delete();

}

### Script

Copy and paste jquery in public/js/jquery.js

Create a script called eliminar.js and within that put

function eliminar(eliminar)

{

$.ajax({

method : 'delete',

url : $(eliminar).attr('data-t'),

data : {

\_token : $(eliminar).data('token')

},

success: function(result) {

// Do something with the result

alert("Registro eliminado");

$(eliminar).parent().parent().remove();

}

});

}

In href of index view there is

<a style="cursor:pointer;" data-t="{{ url('productos/'.$producto->id) }}" data-token="{{ csrf\_token() }}" onclick="eliminar(this)">Eliminar</a>

# Create a rute of specific method of controller

Route::get('photos/popular', 'PhotoController@method');

# Create flash message

In controller put

\Session::flash('flash\_message','Registro guardado');

In teamplate put

@if(session('flash\_message'))

{{session('flash\_message')}}

@endif

# Old values in form

This is just when you send the form and you don’t want that the records will reset

<input type="text" name="name" value="{{ old('name') }}" class="form-control">

THE TYPE SELECT is in this way

<select name="tipo\_documento" required>

<option value="">Seleccione...</option>

<option value="1" @if (old('tipo\_documento') == '1') selected="selected" @endif>Cedula</option>

<option value="2" @if (old('tipo\_documento') == '2') selected="selected" @endif>Rut</option>

<option value="3" @if (old('tipo\_documento') == '3') selected="selected" @endif>Cedula extranjeria</option>

</select>

Remember when is foreach

<select name="curso">

<option value="">Seleccione el curso que desea cursar</option>

@foreach ($cursos as $curso)

<option value="{{ $curso->id }}" @if (old('curso') == $curso->id ) selected="selected" @endif>{{ $curso->fullname }}</option>

@endforeach

</select>

# Testing

First execute this command

composer require laravel/dusk

In App\Providers put into register

public function register()

{

//

if ($this->app->environment('local', 'testing')) {

$this->app->register(DuskServiceProvider::class);

}

}

Next, run the dusk:install Artisan command:

php artisan dusk:install

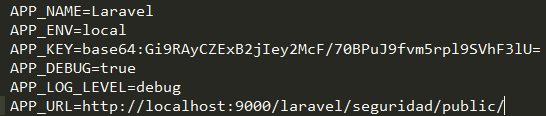
## Generating a File test

Put as you like but remember that the last word must be Test

php artisan dusk:make TestExampleTest

The file will be created in Tests\Browser as TestExampleTest

In the file .ev put in APP\_URL the address that are you using. In this case.



public function testExample()

{

$this->browse(function (Browser $browser) {

$browser->visit('/ingreso')

->type('usuario','santiago5020g@hotmail.com')

->type('clave','1234')

->press('input[type=submit]')

->assertPathIs('/laravel/seguridad/public/usuarios')

->clickLink('Crear Usuario')

->assertPathIs('/laravel/seguridad/public/usuarios/create')

->type('nombre','Rodrigo sa')

->type('email','prueba2@hotmail.com')

->type('telefono','301020')

->type('clave','1234')

->check('div input[name="rol[]"]','1')

->press('input[type=submit]')

->assertPathIs('/laravel/seguridad/public/usuarios')

->assertSee('prueba1@hotmail.com');

});

}