APIAuthemticated V-0.0.1 PHP-BOOTSTRAP 4 -LARAVEL MSQL
Johan Alexander Farfán Sierra
Auxiliar de IT
Bogotá, 26-04-2021

Contenido

APIAuthemticated V-0.0.1 PHP-BOOTSTRAP 4 -LARAVEL MSQL
Johan Alexander Farfán Sierra
Auxiliar de IT
Bogotá, 26-04-2021

DOCUMENTACION TÉCNICA

Instalar el proyecto:

composer create-project --prefer-dist laravel/laravel Nombre_del_proyecto "5.4.*"

Instalar el automatico:

php artisan make:auth

Configuarar la conexción a la base de datos:

DB_CONNECTION=mysql

DB_HOST=127.0.0.1

DB_PORT=3306

DB_DATABASE=apiauthentication

DB_USERNAME=root

DB_PASSWORD=

Crear los modelos:

php artisan make:model "Nombredelmodelo"

Crear los controladores:

php artisan controller Carpeta/Nopmbredelmodelo -r

APIAuthemticated V-0.0.1 PHP-BOOTSTRAP 4 -LARAVEL MSQL
Johan Alexander Farfán Sierra
Auxiliar de IT
Bogotá, 26-04-2021

Implementar los controladores

Controlador Users Para la API:

```
//Validación de acceso por token de verificación
    public function __construct(){

        $this->middleware('auth.basic', ['only'=>['store', 'update', 'destroy']]);

    }
```

```
    //Listar los usuarios
    public function index(Request $request)
    {

        if ($request->has('Buscar')) {

            $users = User::Where('id', 'like', '%' . $request->Buscar. '%')
                ->orWhere('document_number', 'like', '%' . $request->Buscar .'%' )
                ->orWhere('email', 'like', '%' . $request->Buscar .'%' )
                ->orWhere('name', 'like', '%' . $request->Buscar .'%' )
                 ->get();
        } else {
        $users = User::all();

        return response()->json(['data'=> $users], 200);
        }

        return response()->json(['data'=> $users], 200);
    }
```

APIAuthemticated V-0.0.1 PHP-BOOTSTRAP 4 -LARAVEL MSQL
Johan Alexander Farfán Sierra
Auxiliar de IT
Bogotá, 26-04-2021

```php
//Crear los usuarios
   public function store(Request $request)
   {

       try {

           $validators = [

               'name' => 'required',
               'last_name' => 'required',
               'document_number' => 'required',
               'cel_phone' => 'required',
               'email' => 'required|email|unique:users',
               'password'=> 'required|min:6'

           ];

           $this->validate($request, $validators);

           $campos = $request->all();
           $campos['password'] = bcrypt($request->password);
           $campos['verified'] = User::USUARIO_NO_VERIFICADO;
           $campos['verification_token'] = User::generarVerificationToken();

           User::create($campos);

           return response(['message'=> 'El Usuario.' '
           .$request->get('email').' '. 'fue creado exitosamente'], 200);

       } catch (\Throwable $th) {

           return response(['message'=> 'El Usuario.' '
            .$request->get('email').' '. 'Ya existe'], 422);
       }



   }
```

APIAuthemticated V-0.0.1 PHP-BOOTSTRAP 4 -LARAVEL MSQL
Johan Alexander Farfán Sierra
Auxiliar de IT
Bogotá, 26-04-2021

```
 public function show($id)
    {

        $user = User::find($id);

        return response()->json(['data'=> $user], 200);
    }
```

```
//Actualizar usuarios
    public function update(Request $request, $id)

    {
        try {

            $user = User::findOrFail($id);

            if($request->has('name')){
                $user->name = $request->name;
            }
            if($request->has('email') && $user->email !=$request->email){
                $user->verified = User::USUARIO_NO_VERIFICADO;
                $user->verification_token = User::generarVerificationToken();
                $user->email = $request->email;
            }
            if($request->has('password')){
                $user->password = bcrypt($request->password);
            }

            if(!$user->isDirty()){

                return response(['message'=> 'El campo a actualizar
               no debe ser igual al valor actual', 'code'=>422], 422);

            }

            $user->save();
            return response(['data'=> $user], 200);

        } catch (\Throwable $th) {
            return response(['message'=> 'no se pudo actualizar'], 422);
        }
}
```

APIAuthemticated V-0.0.1 PHP-BOOTSTRAP 4 -LARAVEL MSQL
Johan Alexander Farfán Sierra
Auxiliar de IT
Bogotá, 26-04-2021

```
//Eliminar usuarios
    public function destroy($id)
    {

        $user = User::find($id);
        $user->delete();
        return response(['message'=> 'Usuario eliminado exitosamente'], 200);
    }
```

Controlador Users para el FrondTend:

```
//Validación de acceso por token de verificación
    public function __construct()
    {
        $this->middleware('auth.basic', ['only'=>['store', 'update', 'destroy']]);
    }
```

```
//Listar los usuarios
    public function index(Request $request)
    {

        if ($request->has('Buscar')) {

            $users = User::Where('id', 'like', '%' . $request->Buscar. '%')
         ->orWhere('document_number', 'like', '%' . $request->Buscar .'%' )
          ->orWhere('email', 'like', '%' . $request->Buscar .'%' )
          ->orWhere('name', 'like', '%' . $request->Buscar .'%' )
          ->get();
        } else {
        $users = User::all();
        return view('home')->with('users',$users );

        }


    }
```

APIAuthemticated V-0.0.1 PHP-BOOTSTRAP 4 -LARAVEL MSQL
Johan Alexander Farfán Sierra
Auxiliar de IT
Bogotá, 26-04-2021

```php
//Crear los usuarios
    public function store(Request $request)
    {

        try {

            $validators = [

                'name' => 'required',
                'last_name' => 'required',
                'document_number' => 'required',
                'cel_phone' => 'required',
                'email' => 'required|email|unique:users',
                'password'=> 'required|min:6'

            ];

            $this->validate($request, $validators);

            $campos = $request->all();
            $campos['password'] = bcrypt($request->password);
            $campos['verified'] = User::USUARIO_NO_VERIFICADO;
            $campos['verification_token'] = User::generarVerificationToken();

            User::create($campos);

            return back()->with('Listo', ' El Usuario'.' '
            .$request->get('email').' '. 'fue creado exitosamente '
            . 'Por favor activa tu cuenta con el enlace que fue enviedo a tu correo');


        } catch (\Throwable $th) {


            return back()->with('Error', ' El Usuario'.' '
              .$request->get('email').' '. 'Ya existe');
        }



    }
```

APIAuthemticated V-0.0.1 PHP-BOOTSTRAP 4 -LARAVEL MSQL
Johan Alexander Farfán Sierra
Auxiliar de IT
Bogotá, 26-04-2021

```php
public function show($id)
    {

        $user = User::find($id);

        return response()->json(['data'=> $user], 200);
    }
```

```php
    //Actualizar usuarios
    public function update(Request $request, $id)

    {
        try {

            $user = User::findOrFail($id);

            if($request->has('name')){
                $user->name = $request->name;
            }
            if($request->has('email') && $user->email !=$request->email){
                $user->verified = User::USUARIO_NO_VERIFICADO;
                $user->verification_token = User::generarVerificationToken();
                $user->email = $request->email;
            }
            if($request->has('password')){
                $user->password = bcrypt($request->password);
            }

            if(!$user->isDirty()){

                return back()->with('Error',
                ' El campo a actualizar no debe ser igual al valor actual ');


            }

            $user->save();
            return view('home')->with('users',$user );
            return response(['data'=> $user], 200);

            return back()->with('Listo', ' El usuario fue editado exitosamente ');
```

```
        } catch (\Throwable $th) {
            return back()->with('Error', ' El usuario no se pudo editar ');
        }
    }
```

```
//Eliminar usuarios
    public function destroy($id)
    {
       try {
        $user = User::find($id);
        $user->delete();
        return back()->with('Listo', ' Usuario eliminado exitosamente ');
       } catch (\Throwable $th) {
        return back()->with('Error', ' El Usuario no se pudo eliminar ');
       }
```

Crear ruta Users API :

```
Route::resource('users', 'User\UserController');
```

Implementar las propiedades del modelo User:

```
//Para verificar si el usuario es verificado
    const USUARIO_VERIFICADO = '1';
    const USUARIO_NO_VERIFICADO = '0';
```

APIAuthemticated V-0.0.1 PHP-BOOTSTRAP 4 -LARAVEL MSQL
Johan Alexander Farfán Sierra
Auxiliar de IT
Bogotá, 26-04-2021

```php
//atributos del modelo
    protected $fillable = [
        'name',
        'email',
        'password',
        'last_name',
        'document_number',
        'cel_phone',
        'verified',
        'verification_token',

    ];
```

```php
//atrubuto que oculta los atiibutos que estan al momento de convertir
//el modelo en una array formato json

    protected $hidden = [
        'password',
        'remember_token',
        //'verification_token',
    ];
```

```php
//este metodo nos permite generar automaticamente y apartir del modelo
    //el token del usuario
    public function esVerificado(){

        return $this->verified == User::USUARIO_VERIFICADO;
    }
```

```php
//metodo publico estatico que permitira obtener un token de verificacion
    //generado automaticamente
    public static function generarVerificationToken(){
        return str_random(40);
    }
```

APIAuthemticated V-0.0.1 PHP-BOOTSTRAP 4 -LARAVEL MSQL
Johan Alexander Farfán Sierra
Auxiliar de IT
Bogotá, 26-04-2021

Crear la función de validación en los controladores:

API:

```
//controlador de verifiación de usuario
    public function verify($token)
    {
        try {
            $user = User::where('verification_token', $token)->firstOrFail();
            $user->verified = User::USUARIO_VERIFICADO;


            $user->save();


            return view('MessageVerify');
            return response(['message'=> 'Usuario verificado exitosamente'], 200);

        } catch (\Throwable $th) {
            return response(['message'=> 'El usuario ya fue verificado'], 404);
        }

    }
```

FrondTend:

```
//Controlador de verifiación de usuario
    public function verify($token)
    {
        try {
            $user = User::where('verification_token', $token)->firstOrFail();
            $user->verified = User::USUARIO_VERIFICADO;

            $user->save();
            return view('MessageVerify');
            return back()->with('Listo', ' Usuario verificado exitosamente');

        } catch (\Throwable $th) {

            return back()->with('Error', ' El usuario ya fue verificado');
        }

    }
```

APIAuthemticated V-0.0.1 PHP-BOOTSTRAP 4 -LARAVEL MSQL
Johan Alexander Farfán Sierra
Auxiliar de IT
Bogotá, 26-04-2021

Crear las rutas api de la validación:

```
Route::name('verify')
->get('users/verify/{token}', 'User\UserController@verify');
```

Implementar las propiedades de las migraciones.

```
//En esta funcion se especificar las propiedades de las tablas para la migración
    public function up()
    {
        Schema::create('users', function (Blueprint $table) {
            $table->increments('id');
            $table->string('name');
            $table->string('last_name');
            $table->integer('document_number')->unique();
            $table->biginteger('cel_phone');
            $table->string('email')->unique();
            $table->string('password');
            $table->rememberToken();
            $table->string('verified')->default(User::USUARIO_NO_VERIFICADO);
            $table->string('verification_token')->nullable();
            $table->timestamps();
        });
    }
```

Instalar puzzle para el email:

composer require guzzlehttP/guzzle

Crear correo en gmail, y autentiocarlo con mail trap.

https://mailtrap.io/

APIAuthemticated V-0.0.1 PHP-BOOTSTRAP 4 -LARAVEL MSQL
Johan Alexander Farfán Sierra
Auxiliar de IT
Bogotá, 26-04-2021

Actualizar el archivo config-service.php

```
    'mailgun' => [
        'domain' => env('MAILGUN_DOMAIN'),
        'secret' => env('MAILGUN_SECRET'),
    ],

    'ses' => [
        'key' => env('SES_KEY'),
        'secret' => env('SES_SECRET'),
        'region' => 'us-east-1',
    ],

    'sparkpost' => [
        'secret' => env('SPARKPOST_SECRET'),
    ],

    'stripe' => [
        'model' => App\User::class,
        'key' => env('STRIPE_KEY'),
        'secret' => env('STRIPE_SECRET'),
    ],
```

Configurar. dev con las credenciales de acceso a mailtrap:

MAIL_MAILER=smtp

MAIL_HOST=smtp.mailtrap.io

MAIL_PORT=2525

MAIL_USERNAME=e081362926039c

MAIL_PASSWORD=18c76c77f0b984

MAIL_ENCRYPTION=null

MAIL_FROM_ADDRESS=apijohanauthentication@gmail.com

MAIL_FROM_NAME="API Authentication"

PUSHER_APP_ID=

PUSHER_APP_KEY=

PUSHER_APP_SECRET=

APIAuthemticated V-0.0.1 PHP-BOOTSTRAP 4 -LARAVEL MSQL
Johan Alexander Farfán Sierra
Auxiliar de IT
Bogotá, 26-04-2021

Crear el mail:

php artisan make:mail userCreated

Configurar el mail.

```
public $user;

    //aqui se crea el mensaje para la insatncia
    public function __construct(User $user)
    {
        $this->user = $user;
    }

    /**
     * Build the message.
     *
     * @return $this
     */
    public function build()
    {
        return $this->view('email.welcome')
      ->subject('Por favor confirma tu correo electronico');
    }
```

Actualizar app service provider:

```
public function boot()
    {
        Schema::defaultStringLength(191);
        User::created(function($user){
            Mail::to($user)->send(new userCreated($user));

        });
    }
```

APIAuthemticated V-0.0.1 PHP-BOOTSTRAP 4 -LARAVEL MSQL
Johan Alexander Farfán Sierra
Auxiliar de IT
Bogotá, 26-04-2021

Crear en view:

Una carpeta email y en ella el archivo welcome.balde

Configurar el archivo.

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-
beta3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-
eOJMYsd53ii+scO/bJGFsiCZc+5NDVN2yr8+0RDqr0Ql0h+rP48ckxlpbzKgwra6" crossorigin="anonymous">
  </head>
  <body>
    <h1>hello, {{$user->name}}!</h1>

        <div class="container">
            <div class="row">
                <div class="col-md-8 col-md-offset-2">
                    <div class="panel panel-default">
                        <div class="panel-heading"></div>
                            <h4>Thank you for creating an account. please verify it using
                            the following link:<h4>
                            <a class="btn btn-primary btn-link"
                            href="{{route('verify', $user->verification_token)}}">
                            <h1>Verify account</h1></a>
                            <div class="panel-body">
                        </div>
                    </div>
                </div>
            </div>
        </div>
    <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.9.1/dist/umd/popper.min.js"
integrity="sha384-
SR1sx49pcuLnqZUnnPwx6FCym0wLsk5JZuNx2bPPENzswTNFaQU1RDvt3wT4gWFG" crossorigin="anonymous"><
/script>
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-
beta3/dist/js/bootstrap.min.js" integrity="sha384-
j0CNLUeiqtyaRmlzUHCPZ+Gy5fQu0dQ6eZ/xAww941Ai1SxSY+0EQqNXNE6DZiVc" crossorigin="anonymous"><
/script>
  </body>
</html>
```

APIAuthemticated V-0.0.1 PHP-BOOTSTRAP 4 -LARAVEL MSQL
Johan Alexander Farfán Sierra
Auxiliar de IT
Bogotá, 26-04-2021

En view crear el archivo:

MessageVerify.blade

Configurar el archivo:

```
<!DOCTYPE html>
<html lang="{{ app()->getLocale() }}">
<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <!-- CSRF Token -->
    <meta name="csrf-token" content="{{ csrf_token() }}">

    <title>{{ config('app.APIAuthentication', 'API Home') }}</title>

    <!-- Styles -->
    <link href="{{ asset('css/app.css') }}" rel="stylesheet">
    <style>

    .input-font-family{
        color:#212529;
    }

    .buutton-orange{
        background:#fd7e14;
        color:#fff;
        width:auth;
    }
   .panel-orange{
        background:#fd7e14;
        color:#fff;
        height:auth;

    }

    .buutton-link{
        color:#212529;
    }

    .a-titles{
        color:#212529;
    }
```

APIAuthemticated V-0.0.1 PHP-BOOTSTRAP 4 -LARAVEL MSQL
Johan Alexander Farfán Sierra
Auxiliar de IT
Bogotá, 26-04-2021

```
     .i-titles{
        color:#212529;
    }
    .container-page{
        background:#6c757d;
    }

    .container-navbar{
        background:#212529;
        color:#212529;
    }

    .link-navbar{
        color:#212529;
    }

 </style>


</head>


<body>

<nav class="navbar navbar-expand-md navbar-light bg-white shadow-sm">
            <div class="container">
                <a class="navbar-brand" href="{{ url('/') }}">
                    {{ config('app.APIAuthentication', 'APIAuthentication') }}
                </a>
                <button class="navbar-toggler" type="button"
                data-toggle="collapse" data-target="#navbarSupportedContent"
              aria-controls="navbarSupportedContent" aria-expanded="false"
              aria-label="{{ __('Toggle navigation') }}">
                    <span class="navbar-toggler-icon"></span>
                </button>

            </div>
        </nav>

<br></br>
```

APIAuthemticated V-0.0.1 PHP-BOOTSTRAP 4 -LARAVEL MSQL
Johan Alexander Farfán Sierra
Auxiliar de IT
Bogotá, 26-04-2021

```
<div class="container">
    <div class="row justify-content-center">
        <div class="col-md-8">
            <div class="card">
                <div class="panel text-center panel-orange">
                <h3><b>¡Congratulations!</b></h3></div>

                <div class="card-body">


                @if(session()->has('message'))
            <div class="alert {{session('alert') ?? 'alert-info'}}">
                {{ session('message') }}
            </div>
            @endif

            <div class="text-center"><h3><b>User verified successfully</b></h3></div>
                <br></br>
                <div class="col-md-12  text-center">
                <a class="btn  panel-orange " href="{{ route('login') }}">Go tu login</a>
                  </div>

                <br></br>


                </div>
            </div>
        </div>
    </div>
</div>

    <!-- Scripts -->
    <script src="{{ asset('js/app.js') }}"></script>
</body>
</html>
```

APIAuthemticated V-0.0.1 PHP-BOOTSTRAP 4 -LARAVEL MSQL
Johan Alexander Farfán Sierra
Auxiliar de IT
Bogotá, 26-04-2021

Crear las rutas web:

Tener en cuenta que las rutas web están modificadas de acuerdo con las vistas, en este caso pueden ser modificadas de acuerdo con la necesidad del frontend.

```php
//Autorutas de los foormularios AUTH
Auth::routes();

//Ruta para listar reistros
Route::get('/home', 'HomeController@index')->name('home');
///Ruta para agregar regstros
Route::post('/home', 'HomeController@store')->name('home');
//Ruta para eliminar registros
Route::delete('/home/{id}', 'HomeController@destroy')->name('home.destroy');
//Ruta para editar registros
Route::patch('/home/{id}', 'HomeController@update')->name('home.update');
//Ruta para verificar el token
Route::name('MessageVerify')->get('users/verify/{token}', 'HomeController@verify');

//Ruta de redireccionamiento al login
Route::get('/', function () {
    return view('auth.login');
})->middleware('guest');
```

¡Configurar FrondTend a gusto!

Pueden guiarse de los formularios