

```
cd /c/xampp/htdocs/proyectos
```

```
composer create-project laravel/laravel laravelpassport "7.*"
```

```
cd laravelpassport
```

```
$ php artisan --version  
Laravel Framework 7.30.4
```

```
$ php artisan serve  
Laravel development server started: http://127.0.0.1:8000  
[Mon Sep 13 02:24:43 2021] PHP 7.4.12 Development Server (http://127.0.0.1:8000)  
started  
http://127.0.0.1:8000/
```

```
$ code .
```

Teclar Ctrl + C

CONFIGURAR el archivo .env
APP_NAME=Autenticacion
DB_DATABASE=laravelpassport

CONFIGURAR Carpeta config/app.php
Reemplazar 'timezone' => 'UTC',
Por
'timezone' => 'America/Bogota',

Actualizar handler.php

Quedando así:

```
<?php  
  
namespace App\Exceptions;  
  
use Dotenv\Exception\ValidationException;  
use Illuminate\Auth\Access\AuthorizationException;  
use Illuminate\Auth\AuthenticationException;  
use Illuminate\Database\Eloquent\ModelNotFoundException;  
use Illuminate\Database\QueryException;  
use Illuminate\Foundation\Exceptions\Handler as ExceptionHandler;  
use Symfony\Component\HttpKernel\Exception\HttpException;  
use Throwable;  
  
class Handler extends ExceptionHandler  
{  
    /**
```

```

    * A list of the exception types that are not reported.
    *
    * @var array
    */
protected $dontReport = [
    //
];

/**
 * A list of the inputs that are never flashed for validation exceptions
 *
 * @var array
 */
protected $dontFlash = [
    'password',
    'password_confirmation',
];

/**
 * Report or log an exception.
 *
 * @param \Throwable $exception
 * @return void
 *
 * @throws \Throwable
 */
public function report(Throwable $exception)
{
    parent::report($exception);
}

/**
 * Render an exception into an HTTP response.
 *
 * @param \Illuminate\Http\Request $request
 * @param \Throwable $exception
 * @return \Symfony\Component\HttpFoundation\Response
 *
 * @throws \Throwable
 */
public function render($request, Throwable $exception)
{
    if($exception instanceof ModelNotFoundException){

```

```

        return response()-
>json(["res" => false, "error" => "Error de modelo"], 400);
    }

    if($exception instanceof QueryException){
        return response()-
>json(["res" => false, "message" => "Error de consulta BDD " , $exception-
>getMessage()], 500);
    }

    if($exception instanceof HttpException){
        return response()-
>json(["res" => false, "message" => "Error de ruta"], 404);
    }

    if($exception instanceof AuthenticationException){
        return response()-
>json(["res" => false, "message" => "Error de autenticación"], 401);
    }

    if ($exception instanceof AuthorizationException) {
        return response()-
>json(["res" => false, "message" => "Error de autorización, no tiene permiso
s"], 403);
    }

    return parent::render($request, $exception);
}
}

```

Ahora creamos un nuevo controlador, el de usuario
 Michelle@DESKTOP-F3H6SA3 MINGW64 /c/xampp/htdocs/proyectos/ laravelpassportf
 \$ php artisan make:controller **UserController**
 Controller created successfully.

El controlador quedaría así:

```

<?php

namespace App\Http\Controllers;
use App\User;
use Illuminate\Http\Request;

```

```

use Illuminate\Support\Facades\Hash;
use Illuminate\Support\Str;

class UserController extends Controller
{
    public function store(Request $request)
    {
        $input = $request->all();
        $input['password'] = Hash::make($request->password);
        User::create($input);
        return response()->json([
            'res'=>true,
            'message' => 'Usuario Creado Correctamente'
        ]);
    }

    public function login(Request $request)
    {
        $user=User::whereEmail($request->email)->first();
        if(!is_null($user)&&Hash::check($request->password,$user->password))
        {
            $token = $user->createToken('Autenticacion')->accessToken;

            return response()->json([
                'res'=>true,
                'token' =>$token,
                'message' => 'Bienvenido al sistema'
            ]);
        } else {
            return response()->json([
                'res'=>false,
                'message' => 'Cuenta o password incorrecto'
            ]);
        }
    }

    public function logout()
    {
        $user = auth()->user();
        $user->tokens->each(function($token, $key){
            $token->delete();
        });
        $user->save();
        return response()->json(['res' => true,
            'message' => "Adios"]);
    }
}

```

```
}
```

Ahora vamos a registrar la Api, la ruta
`Route::post('users', 'UserController@store');`

`Route::post('login', 'UserController@login');`

Y Logout

Implementar **LARAVEL PASSPORT**

Ejecutar

```
composer require laravel/passport "~9.0"
```

```
php artisan migrate
```

```
php artisan passport:install
```

Agregar en el modelo

```
use Laravel\Passport\HasApiTokens;
```

```
use HasApiTokens, Notifiable;
```

Ir a AuthServiceProvider.php y agregar `Passport::routes();`

E importarlo arriba

```
use Laravel\Passport\Passport;
```

Quedando asi:

```
public function boot()  
{  
    $this->registerPolicies();
```

```
    Passport::routes();

    //
}
```

Ir a `config/auth.php`

Línea 45 y cambiar

```
'driver' => 'token',
```

Por

```
'driver' => 'passport',
```

Ejecutar

```
php artisan passport:client --personal
```

Nos crea esta llave

```
Personal access client created successfully.
Client ID: 3
Client secret: PArknQdu1nA4xAAQPa2LiKXPfjzTE1MvejJG0g9M
```

Ir a `AuthServiceProvider.php` y agregar la siguiente línea debajo de

Debajo

```
Passport::personalAccessClientId
```

Quedando así:

```
<?php

namespace App\Providers;

use Illuminate\Foundation\Support\Providers\AuthServiceProvider as ServiceProvider;
use Illuminate\Support\Facades\Gate;
use Laravel\Passport\Passport;

class AuthServiceProvider extends ServiceProvider
```

```

{
    /**
     * The policy mappings for the application.
     *
     * @var array
     */
    protected $policies = [
        // 'App\Model' => 'App\Policies\ModelPolicy',
    ];

    /**
     * Register any authentication / authorization services.
     *
     * @return void
     */
    public function boot()
    {
        $this->registerPolicies();
        Passport::routes();

        Passport::personalAccessClientId(3);
    }
}

```

Verificar

```

{
    "name": "MARVIN BLAS",
    "email": "marvin.software@gmail.com",
    "password": "12345"
}

```