

CUSTOM GUARD AUTHENTICATION

Mặc định Laravel sẽ sử dụng bảng users để làm chức năng xác thực người dùng.

Trong trường hợp ứng dụng của chúng ta có nhiều đối tượng khác nhau mà vẫn muốn kế thừa phần xác thực có sẵn trong Laravel. Lúc này chúng ta cần phải **Custom Guard**

Trong bài giảng sau, chúng ta sử dụng đối tượng bác sĩ (Doctor)

Tạo Model

File: App\Model\Doctor.php

```
namespace App\Models;

use Illuminate\Support\Facades\Hash;
use Illuminate\Database\Eloquent\Model;
use Illuminate\Notifications\Notifiable;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Foundation\Auth\User as Authenticatable;

class Doctor extends Authenticatable
{
    use HasFactory, Notifiable;

    protected $guarded = [];

    public function setPasswordAttribute($value)
    {
        $this->attributes['password'] =
        Hash::make($value);
    }
}
```

```
public function scopeIsActive($query)
{
    return $query->where('is_active',1);
}
}
```

Tạo Migration

```
public function up()
{
    Schema::create('doctors', function (Blueprint $table) {
        $table->id();
        $table->string('name');
        $table->string('email',32)->unique();
        $table->string('password',255);
        $table->boolean('is_active')->default(true);
        $table->timestamps();
    });
}
```

Tạo Guard

Để tạo thêm Guard, bạn cần vào file `config/auth.php` để cấu hình

```
'guards' => [
    'web' => [
        'driver' => 'session',
        'provider' => 'users',
    ],

    'doctor' => [
        'driver' => 'session',
        'provider' => 'doctors',
    ],
],
```

```
    'api' => [  
        'driver' => 'token',  
        'provider' => 'users',  
        'hash' => false,  
    ],  
],
```

```
'providers' => [  
    'users' => [  
        'driver' => 'eloquent',  
        'model' => App\Models\User::class,  
    ],  
    'doctors' => [  
        'driver' => 'eloquent',  
        'model' => App\Models\Doctor::class,  
    ],  
  
    // 'users' => [  
    //     'driver' => 'database',  
    //     'table' => 'users',  
    // ],  
],
```

```
'passwords' => [  
    'users' => [  
        'provider' => 'users',  
        'table' => 'password_resets',  
        'expire' => 60,  
        'throttle' => 60,  
    ],  
    'doctors' => [  
        'provider' => 'doctors',  
        'table' => 'password_resets',  
        'expire' => 60,  
    ],  
],
```

```
        'throttle' => 60,  
    ],  
],
```

Tạo Route

Tạo file: `routes/doctor.php`

```
use Illuminate\Support\Facades\Auth;  
use Illuminate\Support\Facades\Route;  
use App\Http\Controllers\Doctor\Auth\LoginController;  
  
Route::name('doctor.')->namespace('Doctor')->prefix('doctor')->group(function(){  
  
    Route::namespace('Auth')->middleware('guest:doctor')->group(function(){  
        //login route  
  
        Route::get('/login','LoginController@login')->name('login');  
  
        Route::post('/login','LoginController@processLogin');  
    });  
  
    Route::namespace('Auth')->middleware('auth:doctor')->group(function(){  
  
        Route::post('/logout',function(){  
            Auth::guard('doctor')->logout();  
            return redirect()->action([  
                LoginController::class,
```

```
        'login'
    ]);
    }->name('logout');

});

});
```

Mở file *app/Providers/RouteServiceProvider.php*

```
namespace App\Providers;

use Illuminate\Cache\RateLimiting\Limit;
use
Illuminate\Foundation\Support\Providers\RouteServiceProvi
der as ServiceProvider;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\RateLimiter;
use Illuminate\Support\Facades\Route;

class RouteServiceProvider extends ServiceProvider
{

    public const HOME = '/home';

    public const DOCTOR = '/doctor/home';

    protected $namespace = 'App\\Http\\Controllers';

    public function boot()
    {
        $this->configureRateLimiting();

        $this->routes(function () {
            Route::prefix('api')
```

```

        ->middleware('api')
        ->namespace($this->namespace)
        ->group(base_path('routes/api.php'));

Route::middleware('web')
    ->namespace($this->namespace)
    ->group(base_path('routes/web.php'));

//our custom route path
Route::middleware('web')
    ->namespace($this->namespace)
    ->group(base_path('routes/doctor.php'));
});
}

protected function configureRateLimiting()
{
    RateLimiter::for('api', function (Request
$request) {
        return
Limit::perMinute(60)->by(optional($request->user())->id
?: $request->ip());
    });
}
}

```

Tạo Controller

Tạo file: App\Http\Controllers\Doctor\Auth\LoginController.php

```

namespace App\Http\Controllers\Doctor\Auth;

use Illuminate\Http\Request;
use Facades\App\Helper\Helper;
use App\Http\Requests\LoginRequest;

```

```
use App\Http\Controllers\Controller;
use App\Models\Doctor;
use Illuminate\Support\Facades\Auth;
use Illuminate\Support\Facades\View;
use App\Providers\RouteServiceProvider;
use Symfony\Component\HttpFoundation\Response;

class LoginController extends Controller
{
    public function login()
    {
        if(View::exists('doctor.auth.login'))
        {
            return view('doctor.auth.login');
        }
        abort(Response::HTTP_NOT_FOUND);
    }

    public function processLogin(Request $request)
    {
        $credentials = $request->except(['_token']);

        if(isDoctorActive($request->email))
        {
            if(Auth::guard('doctor')->attempt($credentials))
            {
                return
                redirect(RouteServiceProvider::DOCTOR);
            }
            return redirect()->action([
                LoginController::class,
                'login'
            ])->with('message','Credentials not matced in
our records!');
        }
    }
}
```

```

        return redirect()->action([
            LoginController::class,
            'login'
        ])->with('message','You are not an active
doctors!');
    }
}

```

Tạo Helper

```

use App\Models\Doctor;

if(!function_exists('isDoctorActive'))
{
    function isDoctorActive($email) : bool
    {
        $doctor =
Doctor::whereEmail($email)->IsActive()->exists();

        if($doctor)
        {
            return true;
        }
        return false;
    }
}

```

Tạo view

Tạo file: *resources/views/doctor/auth/login.blade.php*

```

@extends('layouts.app')

@section('content')
<div class="container">

```



```

<div class="row justify-content-center">
  <div class="col-md-8">
    <div class="card">
      <div class="card-header">{{ __( 'Doctor
Login') }}</div>
      @if(session()->has('message'))
        <span>{{ session()->get('message')
}}</span>
      @endif
      <div class="card-body">
        <form method="POST" action="{{
route('doctor.login') }}">
          @csrf

          <div class="form-group row">
            <label for="email"
class="col-md-4 col-form-label text-md-right">{{
__('E-Mail Address') }}</label>

            <div class="col-md-6">
              <input id="email"
type="email" class="form-control @error('email')
is-invalid @enderror" name="email" value="{{ old('email')
}}" required autocomplete="email" autofocus>

              @error('email')
                <span
class="invalid-feedback" role="alert">
                  <strong>{{
$message }}</strong>
                </span>
              @enderror
            </div>
          </div>

          <div class="form-group row">

```

```

<label for="password"
class="col-md-4 col-form-label text-md-right">{{
__('Password') }}</label>

<div class="col-md-6">
    <input id="password"
type="password" class="form-control @error('password')
is-invalid @enderror" name="password" required
autocomplete="current-password">

    @error('password')
    <span
class="invalid-feedback" role="alert">
        <strong>{{
$message }}</strong>
    </span>
    @enderror
</div>
</div>

<div class="form-group row">
    <div class="col-md-6
offset-md-4">
        <div class="form-check">
            <input
class="form-check-input" type="checkbox" name="remember"
id="remember" {{ old('remember') ? 'checked' : '' }}>

            <label
class="form-check-label" for="remember">
                {{ __( 'Remember
Me') }}
            </label>
        </div>
    </div>
</div>
</div>

```

```
<div class="form-group row mb-0">
    <div class="col-md-8
offset-md-4">
        <button type="submit"
class="btn btn-primary">
            {{ __('Login') }}
        </button>

        @if
(Route::has('password.request'))
            <a class="btn
btn-link" href="{{ route('password.request') }}">
                {{ __('Forgot
Your Password?') }}
            </a>
        @endif
    </div>
</div>
</form>
</div>
</div>
</div>
</div>
</div>
@endsection
```

Chức năng Logout

Để gọi phần logout, bạn sử dụng Route và Form như sau

```
<a class="dropdown-item" href="{ route('doctor.logout')
}" onclick="event.preventDefault();
document.getElementById('logout-form').submit();">
  {{ __('Logout') }}
```

```
</a>
```

```
<form id="logout-form" action="{{ route('doctor.logout')  
}}" method="POST" class="d-none">
```

```
    @csrf
```

```
</form>
```