

Tutorial Authorization dengan Gates pada Laravel

 medium.com/dotlocal/belajar-laravel-tutorial-menggunakan-authorization-dengan-gates-2130069bb6d2

May 7, 2020



Belajar **Laravel**

Tutorial dasar Framework Laravel

Oleh : Tobi Ahmad



Pada tutorial kali ini kita akan memakai fitur Authorization menggunakan Gates pada Laravel 7, dengan fitur ini kita bisa membuat role access pada setiap akses user.

Authorization pada project laravel sangat digunakan untuk implementasi auth dan akses user, selain menggunakan middleware, di Laravel juga terdapat Authorization menggunakan Gates dan Policy.

Contoh implementasi Authorization Gates:

- Edit setting harus user dengan roles Admin.
- Update konten harus user yang membuat konten tersebut.

Scope of This Chapter

Pada tutorial ini kita akan membahas tentang fitur Authorization pada framework Laravel, jangan lupa baca tutorial sebelumnya ditautan berikut ini:

Membuat Authentication pada Laravel 7

Pada framework Laravel 7 kita bisa implementasi Auth, seperti membuat fitur login, registrasi yang sudah disediakan...

medium.com

Authorization Gates

Mari kita buat simple project untuk implementasi Authorization Gates

Step 1 — Install & Konfigurasi Laravel

Buat project baru dengan perintah composer berikut pada terminal:

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

Edit konfigurasi database pada file **.env**

```
// edit file .env
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=latihan_blog
DB_USERNAME=root
DB_PASSWORD=password
```

Untuk mempelajari instalasi dan konfigurasi pada Laravel bisa pelajari tautan berikut ini :

Belajar Laravel (Chapter 1 | Instalasi dan Konfigurasi)

Sebelum kita belajar cara menginstal framework Laravel pada laptop kita, kita harus mempersiapkan terlebih dahulu beberapa...

medium.com

Step 2 — Tambah column role pada table users

Pada step ini kita akan membuat migration baru untuk menambahkan column role pada table users. Kita menggunakan datatype enum pada column role dengan value “admin”, “editor”, “author”, dan value “author” sebagai default.

Buka terminal dan jalankan perintah artisan berikut:

```
php artisan make:migration add_role_column_to_users_table
// Created Migration: xx_xx_xx_xx_add_role_column_to_users_table
```

Kemudian edit file migration `xx_xx_xx_xx_add_role_column_to_users_table` yang ada pada folder `database/migrations`:

```

<?phpuse Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;class AddRoleColumnToUsersTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::table('users', function (Blueprint $table) {
            $table->enum('role', ['admin','editor','author'])
                ->default('author');
        });
    }/**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::table('users', function (Blueprint $table) {
            //
        });
    }
}

```

Kemudian jalankan migration dengan perintah artisan berikut pada terminal:

php artisan migrate

```

File Edit View Search Terminal Help

saga@Saga-PC:/var/www/html/latihan/blog$ php artisan make:migration add_role_column_to_users_table
Created Migration: 2020_05_02_082929_add_role_column_to_users_table
saga@Saga-PC:/var/www/html/latihan/blog$ 
saga@Saga-PC:/var/www/html/latihan/blog$ 
saga@Saga-PC:/var/www/html/latihan/blog$ php artisan migrate
Migrating: 2020_05_02_082929_add_role_column_to_users_table
Migrated: 2020_05_02_082929_add_role_column_to_users_table (0.56 seconds)
saga@Saga-PC:/var/www/html/latihan/blog$ 

```

Step 3— Buat Dummy Users dengan Seeder

Buat seeder untuk dummy data users dengan menggunakan perintah artisan **make:seeder**

```
php artisan make:seeder UsersTableSeeder
// Seeder created successfully.
```














































Perintah artisan di atas membuat file baru di **database/seeds/UsersTableSeeder.php**

```
// edit file database/seeds/UsersTableSeeder.php
<?php
use Illuminate\Database\Seeder;
class UsersTableSeeder extends Seeder
{
    /**
     * Run the database seeds.
     *
     * @return void
     */
    public function run()
    {
        for($i=0;$i<15;$i++){
            $input = ['admin','editor','author'];
            $rand_keys = array_rand($input);
            $data[$i] = [
                'name' => $faker->name,
                'email' => $faker->unique()->safeEmail,
                'email_verified_at' => now(),
                'password' => bcrypt('password'),
                'remember_token' => Str::random(10),
                'role' => $input[$rand_keys],
            ];
        }
        DB::table('users')->insert($data);
    }
}
```

Running seeder dengan perintah berikut pada terminal:

```
php artisan db:seed --class=UsersTableSeeder
// Database seeding completed successfully.
```

Silahkan check dummy data yang ada pada database

+ Options											
		id	name	email	role	email_verified_at	password	remember_token	created_at	updated_at	
			1	Prof. Alysson Kunde	lebsack.quinton@example.org	admin	2020-05-02 08:58:19	\$2y\$10\$1raMFD3wUoQxP5AhPBbVO.bhEzmbTj42Dq.cshGue2...	D0XnKeg86m	NULL	NULL
			2	Miss Margarete Kuhn	lwunsch@example.com	admin	2020-05-02 08:58:19	\$2y\$10\$bs/ScTFs4MmkulxMyBSnV.Tr8dVJop1wE7LpNsalcZV...	F0PLPh2ivs	NULL	NULL
			3	Hanna Schaefer	viviane.shields@example.org	admin	2020-05-02 08:58:19	\$2y\$10\$EyrwMaftwGPWFifnSjF5H07HCmZdWwFURmW1schDcjy...	1pe70j4tb0	NULL	NULL
			4	Prof. Jeanie Bednar	vnikolaus@example.org	admin	2020-05-02 08:58:19	\$2y\$10\$03h1N7gPin8JFYMwyKOP3pQgkFYmxAphqr8J9D5O...	Fslcb9Va1C	NULL	NULL
			5	Gilbert Kulas	hills.ramon@example.org	admin	2020-05-02 08:58:19	\$2y\$10\$XM8FKec3zh19x67VfHousODghyujSp956y3GtolEIRy...	XiZewHihRp	NULL	NULL
			6	Royce Konopelski Jr.	lschiller@example.org	admin	2020-05-02 08:58:19	\$2y\$10\$Xs.RZDtbiyh7AQ3POGLLLOb8qJCBP01yoFaBzbYY5vX...	dcyk5ASejt	NULL	NULL
			7	Mr. Walker Langosh II	bmertz@example.com	editor	2020-05-02 08:58:19	\$2y\$10\$6klrBWNobHDNKGbhJ3VxKutPxGXdpG/zeyZaihrOnGA...	V77TKQPFzA	NULL	NULL
			8	Dr. Ubaldo Beier DDS	pmarvin@example.com	admin	2020-05-02 08:58:19	\$2y\$10\$16/IWAv02M0OuOCS90QpYekJijJxTzZ0G9hfyTLyCmD...	Ei9von4NQs	NULL	NULL
			9	Doug Hagenes	catharine78@example.org	admin	2020-05-02 08:58:19	\$2y\$10\$dZ0u9oDenOKPQxY13dZBOG3YLYfupM8KDTeQS4QzC...	XJpcwWPb08	NULL	NULL
			10	Elenor Lebsack	cronin.caitlyn@example.net	admin	2020-05-02 08:58:19	\$2y\$10\$JzZrJf4i4v5m0xIKrFDJ.Uj3vMkSZe/MAWns6mzt/0...	3yK4Mb9ipq	NULL	NULL
			11	Callie Hegmann	myundt@example.org	editor	2020-05-02 08:58:19	\$2y\$10\$CqlDcDdorWHmACbeuPNs7.NILdLE8KxayHEVW1b0alf...	o9U0vKBHh6	NULL	NULL
			12	Terence Nicolas	evelyn.hartmann@example.net	author	2020-05-02 08:58:19	\$2y\$10\$qldvNqAi75Lk/xM7KIX.OQJiHg0cIibGy68BsRMpU...	m70D1ZUbXd	NULL	NULL
			13	Eva Altenwerth	crunolfsson@example.com	admin	2020-05-02 08:58:20	\$2y\$10\$0IzyQ3RZ3hZsGO8AluDa0u1O9an6EvZk7w67ZQ5t0BN...	jqAEb9NKnA	NULL	NULL
			14	Dr. Sammy Romaguera	gfeeney@example.org	editor	2020-05-02 08:58:20	\$2y\$10\$W03W39q/E01LZoKJRXG6be8kWE1qydep7J7IGd72QX4...	BT4365kYA	NULL	NULL
			15	Darion Bogisich	izabella21@example.org	admin	2020-05-02 08:58:20	\$2y\$10\$nh3PgFkRIECXgMbgw9BuSZRSswtQjWkU1Zwhe8Nnf...	A9iMSVpyHY	NULL	NULL

Step 4 — Generate Auth

Install package laravel/ui dengan perintah composer berikut pada terminal:

```
composer require laravel/ui
```

Kemudian untuk membuat view auth dengan bootstrap jalankan perintah **ui** berikut pada terminal:

```
php artisan ui bootstrap --auth
```

Setelah memilih type preset ui yang digunakan, kita perlu menjalankan perintah npm untuk compile view:

Install NPM

```
npm install
```

Run NPM

```
npm run dev
```

Step 5 — Membuat Custom Gates

Untuk membuat gates kita bisa menggunakan fitur facade **Illuminate\Support\Facades\Gate**, kita bisa define custom gate untuk user role access seperti “admin”, “editor”, “author”.

Buka file `app\Providers\AuthServiceProvider.php`

```
<?phpnamespace App\Providers;use Illuminate\Foundation\Support\Providers\AuthServiceProvider as ServiceProvider;use Illuminate\Support\Facades\Gate;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()
    {
        $this->registerPolicies();      Gate::define('isAdmin', function($user) {
            return $user->role == 'admin';
        });      Gate::define('isEditor', function($user) {
            return $user->role == 'manager';
        });      Gate::define('isAuthor', function($user) {
            return $user->role == 'user';
        });      Gate::define('update-post', function ($user, $post) {
            return $user->id === $post->user_id;
        });      }
    }
}
```

Step 6 — Menggunakan Gates Pada Controller

Untuk authorize suatu aksi menggunakan gates bisa menggunakan method **allows** atau **denies**, kita bisa menggunakan kode berikut pada logic yang akan kita buat, misalkan penerapan pada controller:

```
public function update(Request $request)
{
    if (Gate::allows('isAdmin')) {
        // akses logic untuk user dengan role admin
    } else {
        // akses logic untuk user selain role admin
    }
}
public function store(Request $request)
{
    if (Gate::allows('isAuthor')) {
        // akses logic untuk user dengan role author
    } else {
        // akses logic untuk user selain role author
    }
}
public function delete(Request $request)
{
    if (Gate::denies('isAdmin')) {
        // akses logic untuk user selain role admin
    } else {
        // akses logic untuk user dengan role admin
    }
}
```

Pada script di atas facade **Gate** akan menggunakan data user yang sedang login, sedangkan jika kita ingin custom data user pada authorize bisa menggunakan method **forUser** pada facade **Gate**

```
public function updatePost()
{
    $user = User::find(1);
    $post = Post::find(10);    if (Gate::forUser($user)->allows('update-post', $post)) {
        // user dapat mengedit post
    }    if (Gate::forUser($user)->denies('update-post', $post)) {
        // user tidak dapat mengedit
    }
}
```

Authorizing Or Throwing Exceptions

Jika kita ingin menggunakan authorize dan otomatis response error jika define Gate return-nya false, bisa menggunakan method **Gate::authorize**

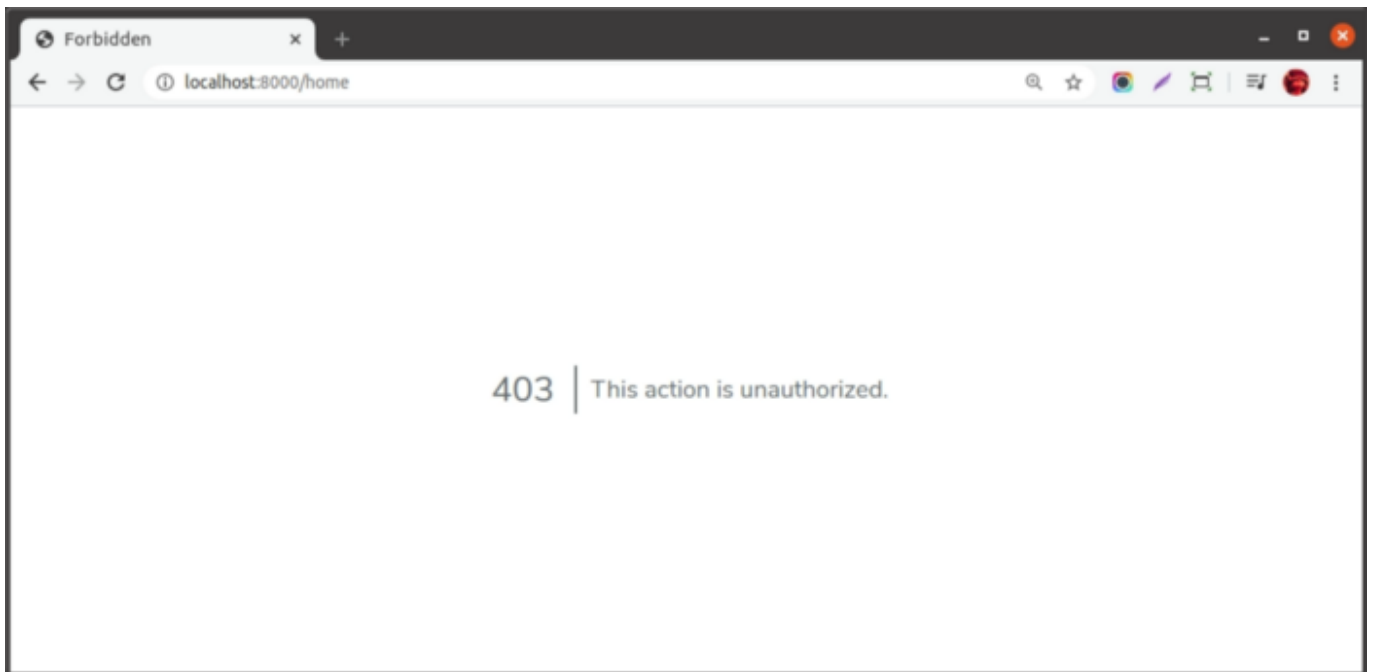
```
public function delete()
{
    $this->authorize('isAdmin');
    // login authorize untuk user role "admin"
}
```

Via Middleware

Untuk menggunakan midlleware pada routes bisa menggunakan method middleware dengan menyisipkan key **can**:

```
// edit file routes/web.phpRoute::get('/home', 'HomeController@index')
->middleware('can:isAdmin')
->name('admin');
```

Jika kita test dengan mengakses halaman home menggunakan user yang menggunakan role “editor”, maka hasilnya error forbidden karena pada route kita telah menambahkan **->middleware('can:isAdmin')**



Step 7— Menggunakan Gates Pada Blade

Kita bisa memanggil Gate dengan menggunakan directive **@can** yang ada pada template blade, edit file blade **home.blade.php**:

body">

Jalankan perintah **php artisan serve** pada terminal, dan akses routes login pada browser, kemudian kita check halaman home.



8/10

Pada fitur login di framework Laravel secara default menggunakan column email dan password. Bagaimana jika ingin...

medium.com

Support the Author

Jika suka dengan artikel ini, bisa support saya ditautan berikut ini

Tobi Ahmad | Karyakarsa

Dukung Tobi Ahmad! Saya adalah seorang web developer di perusahaan yang bergerak di bidang digital dan saya membuat...

karyakarsa.com

Cheers
Tobi Ahmad
Founder of Dot Local

dotlocal

It's time to learn about programming



Written by

Tobi Muhamad

Web and Mobile Developer. Open source enthusiast.

.local

dotlocal

It's time to learn about programming

Learn more.

Medium is an open platform where 170 million readers come to find insightful and dynamic thinking. Here, expert and undiscovered voices alike dive into the heart of any topic and bring new ideas to the surface. [Learn more](#)

Make Medium yours.

Follow the writers, publications, and topics that matter to you, and you'll see them on your homepage and in your inbox. [Explore](#)

Write a story on Medium.

If you have a story to tell, knowledge to share, or a perspective to offer — welcome home. It's easy and free to post your thinking on any topic. [Start a blog](#)



[About](#)

[Write](#)

[Help](#)

[Legal](#)