# Phân quyền người dùng với Laravel Authorization

**Laravel Framework** hỗ trợ sẵn cho lập trình viên việc xây dựng chức năng phân quyền người dùng thông qua Laravel Gate và Laravel Policy

Việc sử Gate hay Policy tùy theo nghiệp vụ của ứng dụng, nhưng phần lớn các ứng dụng đều sử dụng cả Gate và Policy

## Laravel Gate

- Gate là một Closures, nó định nghĩa cho phép/từ chối thực hiện một hành động cụ thể nào đó hay không của một user
- Gate sẽ không gắn liên quan đến model nào cả, Gate sẽ chỉ sử dụng thông tin được cung cấp bởi những tham số
- Gate thường sử dụng ở tầng Controller, để định nghĩa hành động của user là có được phép hay không
- Gate được viết ở **App\Providers\AuthServiceProvider** và sử dụng Gate Facade

### Cú pháp định nghĩa Gate dùng Closure

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

    Gate::define('update-post', function ($user, $post) {
        return $user->id == $post->user_id;
    });
}
```

**Cú pháp định nghĩa dùng Callback Array**

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

    Gate::define('update-post', [Post::class, 'update']);
}
```

**Kiểm tra phân quyền**

```
//Kiểm tra được phép
if (Gate::allows('update-post', $post)) {
    // The current user can update the post...
}

//Kiểm tra không được phép
if (Gate::denies('update-post', $post)) {
    // The current user can't update the post...
}
```

Nếu muốn kiểm tra quyền của 1 user nào đó, sử dụng cú pháp sau:

```
if (Gate::forUser($user)->allows('update-post', $post)) {
    // The user can update the post...
}

if (Gate::forUser($user)->denies('update-post', $post)) {
    // The user can't update the post...
}
```

Nếu muốn kiểm tra quyền bằng **Middleware**, sử dụng cú pháp sau:

```
Route::put('/post/{post}', function (Post $post) {
    // The current user may update the post...
})->middleware('can:update,post');
```

hoặc bạn dùng thông qua phương thức `can()` với cú pháp sau:

```
Route::put('/post/{post}', function (Post $post) {
    // The current user may update the post...
})->can('update', 'post');
```

Nếu muốn kiểm tra quyền thông qua Blade Template, sử dụng cú pháp sau:

```
@can('update', $post)
    <!-- The current user can update the post... -->
@else
    <!-- ... -->
@endcan
```

## Laravel Policy

- Policy là các class quản lý logic trong phân quyền liên quan đến một Model hoặc tài nguyên nào đó
- Trong Policy, Model User cung cấp 2 phương thức hữu ích cho việc xác thực là `can()` và `cant()`
- Phương thức `can()` xác nhận việc user có thể thực hiện một hành động nào đó, trong khi phương thức `cant()`, xác nhận việc user không thể thực hiện hành động được chỉ định
- Policy thường được sử dụng với các hành động CRUD của một model cụ thể

## Tạo Policy

Để tạo Policy mới, hãy sử dụng câu lệnh artisan dưới đây, Laravel sẽ tự động tạo:

```
php artisan make:policy PostPolicy
```

Nếu muốn sinh ra Policy CURD, cần thêm `-model` vào câu lệnh artisan

```
php artisan make:policy PostPolicy --model=Post
```

### Đăng ký Policy

Để sử dụng Policy chúng ta cần đăng ký, để đăng ký chỉ cần bổ sung thuộc tính `$policies` vào `AuthServiceProvider`

```
protected $policies = [
    Post::class => PostPolicy::class,
];

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

    //
}
```

## Logic nghiệp vụ trong Policy

Khi Policy được đăng ký, bạn có thể thêm các phương thức cho mỗi hành động cần cấp quyền

```
class PostPolicy
{
    /**
     * Determine if the given post can be updated by the
user.
     *
     * @param  \App\User  $user
     * @param  \App\Post  $post
     * @return bool
     */
    public function update(User $user, Post $post)
    {
        return $user->id === $post->user_id;
    }
}
```

## Policy Filter

Nếu bạn muốn lọc quyền thực hiện cho 1 Policy, bạn chỉ cần thêm phương thức before() như sau:

```
public function before($user, $ability)
{
     //Chỉ superadmin được phép thực hiện
    if ($user->isSuperAdmin()) {
        return true;
    }
}
```

## Kiểm tra phân quyền Policy

Trong Model User, bạn dùng phương thức `can()` để kiểm tra cho phép và `cant()` để kiểm tra không được phép

```
if ($user->can('update', $post)) {
    //Được phép update post
}

if ($user->cant('update', $post)) {
    //Không được phép update post
}
```

Nếu trong Controller, sử dụng phương thức `authorize()`

```
public function update(Request $request, Post $post)
{
    $this->authorize('update', $post);

    // The current user can update the blog post...
}
```

Ngoài ra, chúng ta có thể sử dụng Middleware, Blade Template giống như **Laravel Gate**