Creating a Role-Specific Dashboard in Laravel 12

In many applications, different users need different dashboards. For example, an **admin** might see system statistics, a **manager** might see team performance, while a **user** only sees their own activity. In **Laravel 12**, you can create **role-specific dashboards** using the Spatie Permissions package and some conditional logic.

In this guide, we'll build dashboards that automatically adapt based on the logged-in user's role. We'll secure routes, load different views, and provide a clean UI for each role.

1 – Defining Roles

Before building dashboards, make sure your roles are defined with Spatie Permissions. For example:

```
php artisan tinker

>>> Role::create(['name' => 'admin']);
>>> Role::create(['name' => 'manager']);
>>> Role::create(['name' => 'user']);Code language: CSS (css)
```

Assign roles to users when registering or via an admin UI. Example:

```
$user->assignRole('admin');Code language: PHP (php)
```

2 – Route Setup

We'll use a single dashboard route that points to a controller. Middleware ensures only authenticated users can access it.

```
// routes/web.php
use App\Http\Controllers\DashboardController;

Route::middleware(['auth'])->get('/dashboard',
[DashboardController::class, 'index'])->name('dashboard');Code language: PHP (php)
```

3 – Dashboard Controller

The controller checks the user's role and returns the correct view. This keeps logic centralized.

```
// app/Http/Controllers/DashboardController.php
namespace App\Http\Controllers;

use Illuminate\Support\Facades\Auth;

class DashboardController extends Controller
{
    public function index()
    {
        $user = Auth::user();

        if ($user->hasRole('admin')) {
            return view('dashboards.admin');
        }

        if ($user->hasRole('manager')) {
```

```
            return view('dashboards.manager');
        }

        return view('dashboards.user');
    }
}Code language: PHP (php)
```

Now, each role gets its own dashboard view file.

4 – Creating Dashboard Views

Let's create separate views for each role:

```
// resources/views/dashboards/admin.blade.php
@extends('layouts.app')

@section('content')
<div class="container">
  <h1>Admin Dashboard</h1>
  <p>System statistics and user management tools go here.</p>
</div>
@endsectionCode language: HTML, XML (xml)
```

```
// resources/views/dashboards/manager.blade.php
@extends('layouts.app')

@section('content')
<div class="container">
  <h1>Manager Dashboard</h1>
  <p>Team performance metrics and project controls go here.</p>
</div>
@endsectionCode language: HTML, XML (xml)
```

```
// resources/views/dashboards/user.blade.php
@extends('layouts.app')

@section('content')
<div class="container">
  <h1>User Dashboard</h1>
  <p>Your personal activity, notifications, and profile tools go
here.</p>
</div>
@endsectionCode language: HTML, XML (xml)
```

Each dashboard is independent, so you can customize the content for the needs of each role.

5 – Securing Role-Specific Pages

Sometimes you'll want to restrict not just dashboards but also subpages. You can use middleware or Blade directives for this.

```
// Example: restrict reports route to managers
Route::middleware(['auth','role:manager'])->get('/reports', function
() {
    return 'Reports Page';
});Code language: PHP (php)
```

```
// Example in Blade
@role('admin')
  <a href="/admin/settings">Settings</a>
@endroleCode language: HTML, XML (xml)
```

This ensures that only users with the correct roles see links and access routes.

Wrapping Up

We created a **role-specific dashboard system in Laravel 12**. Each user sees a different dashboard based on their role, managed with Spatie Permissions. Routes are protected with middleware, and Blade directives ensure the right UI is displayed. This approach provides a tailored experience for each type of user while keeping security intact.

What’s Next

- Building a Team Management System with Laravel Roles & Permissions
- How to Give and Revoke Permissions to Users in Laravel
- Laravel Roles vs Policies: Which One Should You Use?