01_Laravel_Advanced_Routing.md

Implement Dynamic Route Model Binding with Custom Keys and Route Groups with Middleware in Laravel

Objective

Learn how to:

1. Use **custom keys** in **Route Model Binding** (e.g., `slug` instead of `id`).
2. Group routes with **middleware** for authentication, rate limiting, etc.

---

1. Dynamic Route Model Binding with Custom Keys

By default, Laravel resolves Eloquent models using `id`. To use a different column (e.g., `slug`), modify:

Option 1: Explicit Binding in `RouteServiceProvider`

```php
// app/Providers/RouteServiceProvider.php
public function boot()
{
    parent::boot();

    // Bind 'post' parameter to Post model using 'slug' instead of 'id'
    Route::model('post', \App\Models\Post::class);
    Route::bind('post', function ($slug) {
        return \App\Models\Post::where('slug', $slug)->firstOrFail();
    });
}
```

PROF

Option 2: Override `getRouteKeyName()` in the Model

```php
// app/Models/Post.php
class Post extends Model
{
    public function getRouteKeyName()
    {
        return 'slug'; // Use 'slug' instead of 'id' for URLs
    }
}
```

Usage in Routes

```
// routes/web.php
Route::get('/posts/{post}', function (Post $post) {
    return view('posts.show', compact('post'));
});
```

- Now `/posts/my-post-slug` will fetch the post where `slug = 'my-post-slug'`.

---

2. Route Groups with Middleware

Group routes to apply **middleware** (e.g., `auth`, `throttle`, custom middleware).

Example: Admin Routes with Auth & Custom Middleware

```
// routes/web.php
Route::middleware(['auth', 'admin'])->prefix('admin')->group(function ()
{
    Route::get('/dashboard', [AdminController::class, 'dashboard']);
    Route::get('/users', [AdminController::class, 'users']);
});
```

- **`auth`**: Ensures the user is logged in.
- **`admin`**: A custom middleware (e.g., checks if `user->is_admin = true`).
- **`prefix('admin')`**: Prepends `/admin` to all routes in the group.

Example: API Routes with Rate Limiting

```
// routes/api.php
Route::middleware('throttle:60,1')->group(function () {
    Route::get('/posts', [PostController::class, 'index']);
    Route::get('/posts/{post}', [PostController::class, 'show']);
});
```

PROF

- **`throttle:60,1`**: Limits to 60 requests per minute per IP.

Final Implementation

1. Custom Route Binding + Middleware Group

```
// routes/web.php
Route::middleware('auth')->group(function () {
    Route::get('/profile', [ProfileController::class, 'show']);
    // Uses 'slug' instead of 'id' due to RouteServiceProvider or Model
setup
    Route::get('/posts/{post}', [PostController::class, 'show']);
});
```

2. Testing

- Visit `/posts/my-post-slug` → Should fetch the correct post.
- Visit `/admin/dashboard` → Should redirect if not logged in or not an admin.

Key Takeaways

✅ **Custom Route Binding**: Fetch models using `slug`, `username`, etc., instead of `id`.
✅ **Route Groups**: Apply middleware, prefixes, and namespaces to multiple routes efficiently.
✅ **Middleware**: Restrict access (e.g., `auth`, `admin`, `throttle`).

Next Step: Try implementing **Laravel Policies & Gates** for fine-grained authorization! 🚀