



## Structured Notes on Laravel Routing (From the Lecture)

### 1. What is Routing?

- Routes define **URLs** in your app and tell Laravel **how to respond**.
  - Located in routes/web.php.
- 

### 2. Basic Route Syntax

php

CopyEdit

```
Route::get('/', function () {  
    return view('welcome');  
});
```

- Route::get: responds to **GET** requests.
  - '/': the home URL.
  - function: the logic executed when route is visited.
  - view('welcome'): returns the Blade template from resources/views/welcome.blade.php.
- 

### 3. Route with Controller

php

CopyEdit

```
Route::get('/home', [HomeController::class, 'index']);
```

- Links URL /home to HomeController@index.
  - Organizes logic inside controllers under app/Http/Controllers.
- 

### 4. Passing Data to Views

php

CopyEdit

```
Route::get('/about', function () {  
    return view('about', ['name' => 'Pranay']);  
});
```

- Sends data using an array passed to view().
- In Blade, use {{ \$name }} to display it.

---

## 5. Dynamic Route Parameters

php

CopyEdit

```
Route::get('/user/{id}', function ($id) {  
    return "User ID: $id";  
});
```

- {id} is a **placeholder**.
- Incoming URL like /user/5 → \$id = 5.

---

## 6. Optional Parameters

php

CopyEdit

```
Route::get('/post/{id?}', function ($id = null) {  
    return $id ? "Post $id" : "No ID Provided";  
});
```

- ? makes the route parameter optional.
- Provide a **default value** in the function.

---

## 7. Named Routes

php

CopyEdit

```
Route::get('/dashboard', [DashboardController::class, 'show'])  
->name('dashboard');
```

- Helps generate URLs easily: route('dashboard').

---

## 8. Route Groups & Middleware

php

CopyEdit

```
Route::middleware(['auth'])->group(function () {  
    Route::get('/profile', [ProfileController::class, 'show']);  
});
```

```
Route::get('/settings', [SettingsController::class, 'edit']);  
});
```

- Applies shared logic like auth to multiple routes in a group.
- 

## 9. Route::view shortcut

php

CopyEdit

```
Route::view('/contact', 'contact');
```

- A shortcut for simple GET + view routes (no controller needed).
- 

## 10. Best Practices Recap

- 📁 **Single responsibility:** Use controllers for complex logic.
  - 🎯 **Use named routes** for clean URL management.
  - 🛡️ **Use middleware** to secure route groups.
  - 🧩 **Use view()** for simple routes with just a view.
  - 📦 **Dynamic routes** are great for parameters like IDs.
- 

## 🧠 Memory Aids

### ♦ MVC Flow with Routing

Routes → Controller → Model → View

### ♦ Shortcut guide:

- **static page?** → Route::view(...)
  - **Dynamic content?** → Route → Controller
  - **Simple closure & data?** → use Route::get(...)
- 

## ✅ Your Next Step

1. Practice creating routes in routes/web.php.
2. Use php artisan serve to test in browser.
3. Link to controllers and pass data to views.
4. Try dynamic & named routes.
5. Learn about middleware to protect routes later.

