## Adding Meta Tags and Open Graph Data Dynamically in Laravel

# Adding Meta Tags and Open Graph Data Dynamically in Laravel

Meta tags, Open Graph, and Twitter Cards help search engines and social networks understand your content. In Laravel, you can dynamically generate these tags per page, pulling from your database or model attributes. This guide shows how to add SEO meta, Open Graph, and Twitter tags in controllers, Blade templates, and admin forms.

## Controller-Driven Dynamic Meta Tags

For each resource (like posts), you can pass meta data from the controller into the view. If a post has a `meta_title` and `meta_description` column, use them to build SEO-friendly tags.

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

use App\Models\Post;

class PostController extends Controller
{
    public function show(Post $post)
    {
        return view('posts.show', [
            'post' => $post,
```

```
            'meta' => [
                'title' => $post->meta_title ?? $post->title,
                'description' => $post->meta_description ??
str($post->content)->limit(160),
                'image' => $post->cover_image,
            ]
        ]);
    }
}Code language: PHP (php)
```

This way, the Blade view can render correct tags for each post, falling back to defaults when needed.

## Blade Example: Meta + Open Graph + Twitter

Inside your layout Blade, add a `@yield('meta')` section so each page can define its own tags. Here's an example:

```
<!-- resources/views/layouts/app.blade.php -->
<head>
  <title>@yield('title', 'My Blog')</title>
  @yield('meta')
</head>Code language: PHP (php)
```

```
<!-- resources/views/posts/show.blade.php -->
@extends('layouts.app')

@section('title', $meta['title'])

@section('meta')
  <meta name="description" content="{{ $meta['description'] }}">

  <!-- Open Graph -->
```

```
  <meta property="og:title" content="{{ $meta['title'] }}">
  <meta property="og:description" content="{{ $meta['description']
}}">
  <meta property="og:type" content="article">
  <meta property="og:url" content="{{ url()->current() }}">
  <meta property="og:image" content="{{
asset('storage/'.$meta['image']) }}">

  <!-- Twitter Cards -->
  <meta name="twitter:card" content="summary_large_image">
  <meta name="twitter:title" content="{{ $meta['title'] }}">
  <meta name="twitter:description" content="{{ $meta['description']
}}">
  <meta name="twitter:image" content="{{
asset('storage/'.$meta['image']) }}">
@endsection

@section('content')
  <h1>{{ $post->title }}</h1>
  <p>{{ $post->content }}</p>
@endsectionCode language: PHP (php)
```

This setup ensures each post page has correct SEO meta, Open Graph for Facebook/LinkedIn, and Twitter card tags for rich previews.

## UI Example: Admin Form for Meta Fields

Allow editors to define meta title, description, and OG image directly in the admin panel. Add these fields in your post form:

```
<form action="{{ route('posts.store') }}" method="POST"
enctype="multipart/form-data">
  @csrf
```

```
  <label>Title</label>
  <input type="text" name="title">

  <label>Meta Title</label>
  <input type="text" name="meta_title">

  <label>Meta Description</label>
  <textarea name="meta_description"></textarea>

  <label>OG/Twitter Image</label>
  <input type="file" name="cover_image">

  <button type="submit">Save</button>
</form>Code language: PHP (php)
```

These values are then stored in your `posts` table and rendered dynamically in the head tags.

## Global Defaults with Middleware or Service

To avoid missing tags on pages without meta fields, create a middleware or service that sets global defaults.

```
// app/Http/Middleware/DefaultMeta.php
namespace App\Http\Middleware;

use Closure;

class DefaultMeta
{
    public function handle($request, Closure $next)
    {
        view()->share('defaultMeta', [
```

```
            'title' => 'My Laravel Blog',
            'description' => 'Latest Laravel tutorials and guides',
            'image' => '/images/default-og.png',
        ]);

        return $next($request);
    }
}Code language: PHP (php)
```

Register this middleware globally, then in your views use `$meta ?? $defaultMeta` to ensure every page has valid meta tags.

## Wrapping Up

Dynamic meta tags let you optimize every page for SEO and social sharing. With controller-driven meta, Blade templates for OG and Twitter cards, admin UI fields, and global defaults, your Laravel app is fully SEO-ready and shares beautifully on social media.

## What's Next

Keep building your SEO toolkit with these related guides:

- Laravel SEO Guide: Optimizing Meta, Slugs, and Sitemaps
- How to Generate SEO-Friendly URLs and Slugs in Laravel
- How to Build an XML Sitemap Generator in Laravel