

Meta Tags for SEO Performance

Dardan Ternava
Grok

May 24, 2025

1 From the Author

Hi there from Dardan. I wrote this document on Meta-Tags to help with optimising your, or your client's Website with ranking on Search Engines (Google, Bing etc.). While there are more practices to help with SEO ranking, I focused more on the programming aspect of it - The Meta Tags. The idea came to me while i was working on my React website and needed to optimise it for SEO (using react-helmet-async). Grok was a great help on putting this together and there is no need on hiding AI-s help with these kind of projects. I hope you find it helpful even though it is just a small document.

2 Introduction

Adding meta tags to your website's HTML `<head>` section is crucial for improving Search Engine Optimization (SEO) performance. Meta tags provide search engines with information about your page's content, structure, and relevance, influencing how your site is indexed and ranked. This document outlines essential and additional meta tags to enhance your website's SEO, tailored for use with `react-helmet-async` in a React environment.

3 Essential Meta Tags for SEO

These core meta tags should be included on every page.

- **Title Tag**

Purpose: Defines the title of the page, displayed in search engine results and browser tabs.

Best Practices: Keep it between 50-60 characters, include primary keywords, and make it descriptive.

Example

(using -

```
react-helmet-async): import { Helmet } from 'react-helmet-async';
```

```
function HomePage() {
```

```
    return (  
      <div>  
        <Helmet>  
          <title>Home - My Awesome Website</title>  
        </Helmet>  
        <h1>Welcome to My Site</h1>  
      </div>  
    );  
  }
```

• Meta Description

Purpose: Provides a summary of the page content, often shown in search results.

Best Practices: Keep it between 150-160 characters, include keywords, and make it compelling.

Example: <Helmet>
 <meta name="description"
 content="Explore our amazing products and services at
 My Awesome Website. Shop now for the best deals!" />
</Helmet>

• Viewport Meta Tag

Purpose: Ensures the page is responsive and scales correctly on mobile devices, impacting mobile SEO.

Best Practices: Use width=device-width and initial-scale=1.

Example: <Helmet>
 <meta name="viewport" content="width=device-width, initial-scale=1" />
</Helmet>

• Charset

Purpose: Specifies the character encoding for the page, ensuring proper text rendering.

Best Practices: Use UTF-8 for broad compatibility.

Example: <Helmet>
 <meta charset="UTF-8" />
</Helmet>

4 Open Graph Meta Tags (for Social Media Sharing)

Open Graph (OG) tags control how your page appears when shared on platforms like Facebook, Twitter, and LinkedIn.

• og:title

Purpose: Defines the title for social media sharing.

Example: <Helmet>

```
<meta property="og:title" content="Home - My Awesome Website" />
</Helmet>
```

- **og:description**

Purpose: Provides a description for social sharing.

Example: <Helmet>

```
<meta property="og:description"
  content="Explore our amazing products and services at My Awesome Website" />
</Helmet>
```

- **og:image**

Purpose: Specifies an image to display when the page is shared.

Best Practices: Use an image at least 1200x630 pixels for best results.

Example: <Helmet>

```
<meta property="og:image"
  content="https://myawesomewebsite.com/images/og-image.jpg" />
</Helmet>
```

- **og:url**

Purpose: Defines the canonical URL of the page for sharing.

Example: <Helmet>

```
<meta property="og:url" content="https://myawesomewebsite.com/" />
</Helmet>
```

- **og:type**

Purpose: Indicates the type of content (e.g., website, article).

Best Practices: Use **website** for homepages, **article** for blog posts.

Example: <Helmet>

```
<meta property="og:type" content="website" />
</Helmet>
```

- **og:site name**

Purpose: Specifies the name of your website.

Example: <Helmet>

```
<meta property="og:site_name" content="My Awesome Website" />
</Helmet>
```

5 Twitter Card Meta Tags

Twitter uses its own set of meta tags for sharing, complementing Open Graph tags.

- **twitter:card**

Purpose: Defines the card type (e.g., summary, summary large image).

Best Practices: Use `summary large image` for images, `summary` for text.

Example:

```
<Helmet>
  <meta name="twitter:card" content="summary_large_image" />
</Helmet>
```

- **twitter:title**

Purpose: Sets the title for Twitter sharing.

Example:

```
<Helmet>
  <meta name="twitter:title" content="Home - My Awesome Website" />
</Helmet>
```

- **twitter:description**

Purpose: Provides a description for Twitter.

Example:

```
<Helmet>
  <meta name="twitter:description"
    content="Explore our amazing products." />
</Helmet>
```

- **twitter:image**

Purpose: Specifies an image for Twitter cards.

Example:

```
<Helmet>
  <meta name="twitter:image"
    content="https://myawesomewebsite.com/images/twitter-image.jpg" />
</Helmet>
```

- **twitter:site (Optional)**

Purpose: Your Twitter handle for the site (e.g., @username).

Example:

```
<Helmet>
  <meta name="twitter:site" content="@MyAwesomeSite" />
</Helmet>
```

6 Additional SEO Meta Tags

These tags provide extra information to search engines and enhance indexing.

- **robots**

Purpose: Instructs search engine bots how to crawl and index the page.

Best Practices: Use `index`, `follow` for most pages, `noindex`, `nofollow` for private pages.

Example: `<Helmet>`

```
<meta name="robots" content="index, follow" />
</Helmet>
```

- **canonical**

Purpose: Specifies the preferred URL to avoid duplicate content issues.

Example: `<Helmet>`

```
<link rel="canonical" href="https://myawesomewebsite.com/" />
</Helmet>
```

- **keywords (Optional)**

Purpose: Lists relevant keywords for the page (less impactful today but used by some search engines).

Best Practices: Keep it concise, 5-10 keywords.

Example: `<Helmet>`

```
<meta name="keywords" content="products, services, shopping, deals" />
</Helmet>
```

- **author (Optional)**

Purpose: Indicates the pages author (can help with authorship attribution).

Example: `<Helmet>`

```
<meta name="author" content="John Doe" />
</Helmet>
```

- **application-name (Optional)**

Purpose: Specifies the name of the web application (useful for PWAs).

Example: `<Helmet>`

```
<meta name="application-name" content="My Awesome Website" />
</Helmet>
```

7 Structured Data Meta Tags (JSON-LD)

Structured data helps search engines understand your content better, potentially leading to rich snippets.

- **JSON-LD Script**

Purpose: Provides structured data (e.g., for products, articles, reviews).

Best Practices: Use Schema.org vocabulary and validate with Googles Structured Data Testing Tool.

Example (webpage): <Helmet>

```
<script type="application/ld+json">
{
  "@context": "https://schema.org",
  "@type": "WebPage",
  "name": "Home",
  "description": "Explore our amazing products and services.",
  "url": "https://myawesomewebsite.com/"
}
</script>
</Helmet>
```

Example (product): <Helmet>

```
<script type="application/ld+json">
{
  "@context": "https://schema.org",
  "@type": "Product",
  "name": "Awesome Hoodie",
  "image": "https://myawesomewebsite.com/images/hoodie.jpg",
  "description": "A comfy hoodie for all seasons.",
  "sku": "12345",
  "offers": {
    "@type": "Offer",
    "priceCurrency": "USD",
    "price": "29.99"
  }
}
</script>
</Helmet>
```

8 Notes

- **Dynamic Content:** With `react-helmet-async`, set these tags dynamically based on the page or component using props or state.
- **Testing:** Use tools like Googles Rich Results Test, Facebook Sharing Debugger, and Twitter Card Validator to ensure correct interpretation.

- **Avoid Overstuffing:** Avoid unnecessary tags or duplicate content to prevent confusion or penalties.
- **Compatibility:** Monitor for runtime issues with `react-helmet-async` and `react@19.1.0`, especially after using `-force` for installation.

9 Example Combined Implementation

Below is an example combining these tags in a React component:

```
import { Helmet } from 'react-helmet-async';

function HomePage() {
  return (
    <div>
      <Helmet>
        <title>Home - My Awesome Website</title>
        <meta charset="UTF-8" />
        <meta name="viewport" content="width=device-width,
          initial-scale=1" />
        <meta name="description"
          content="Explore our amazing products and services." />
        <meta name="robots" content="index, follow" />
        <link rel="canonical" href="https://myawesomewebsite.com/" />
        <meta property="og:title"
          content="Home - My Awesome Website" />
        <meta property="og:description"
          content="Explore our amazing products." />
        <meta property="og:image"
          content="https://myawesomewebsite.com/images/og-image.jpg" />
        <meta property="og:url" content="https://myawesomewebsite.com/" />
        <meta property="og:type" content="website" />
        <meta property="og:site_name" content="My Awesome Website" />
        <meta name="twitter:card" content="summary_large_image" />
        <meta name="twitter:title" content="Home - My Awesome Website" />
        <meta name="twitter:description" content="Explore our amazing products." />
        <meta name="twitter:image"
          content="https://myawesomewebsite.com/images/twitter-image.jpg" />
        <script type="application/ld+json">
{
  "@context": "https://schema.org",
  "@type": "WebPage",
  "name": "Home",
  "description": "Explore our amazing products and services.",
  "url": "https://myawesomewebsite.com/"
}
        </script>
      </Helmet>
    </div>
  );
}
```

```
        <h1>Welcome to My Site</h1>
      </div>
    );
}

export default HomePage;
```