



A Website Without Meta Tags Is Like A Shop Without A Signboard



About Me

- 3+ years of experience in PHP, Drupal, and web development
- Drupal Engineer at Axelerant
- Worked across multiple projects like multilingual, Migration, and Media & Publishing
- Part of the Volunteers team in DrupalCamp Mumbai 2019
- I like to travel and explore new places like mountains.



Axelerant 2021 - Till Now

 @shani-maurya

 @Er. Shani Maurya

 @shaniku98





Agenda

- An overview of the meta tag module and its features.
- How to manage it across the entire website?
- How meta tags can help generate the Twitter Card Meta Tags?
- How to configure the Open Graph Meta Tag for Facebook?
- Analyzing if it's possible to customize the value of a meta tag programmatically.



Metatag Modules



Drupal provides the metatag module as a contributed module. It allows users to configure structured metadata such as meta descriptions and meta keywords for a Drupal website.

This module also provides support for a wide range of Drupal meta tags such as:

- Open Graph Protocol
- Twitter Cards and Facebook
- Favicons and Mobile & UI Adjustments
- Pinterest and Google Plus
- Views



^ SEO	
<input type="checkbox"/>	Metatag Manage meta tags for all entities.
<input type="checkbox"/>	Metatag Custom Routes (Paths) Allows assigning meta tags to be used on custom routes, equivalent to custom paths.
<input type="checkbox"/>	Metatag Extended Permissions Adds individual permissions for each meta tag, allowing for fine-grained access to the meta tags. Note: this may lead to performance issues on the permissions admin page, please see the included README.txt file for details.
<input type="checkbox"/>	Metatag: App Links Provides support for applinks.org meta tags.
<input type="checkbox"/>	Metatag: Dublin Core Provides the fifteen Dublin Core Metadata Element Set 1.1 meta tags from the Dublin Core Metadata Institute .
<input type="checkbox"/>	Metatag: Dublin Core Advanced Provides forty additional meta tags from the Dublin Core Metadata Institute .
<input type="checkbox"/>	Metatag: Facebook A set of meta tags specially for controlling advanced functionality with Facebook.
<input type="checkbox"/>	Metatag: Favicons Provides support for many different favicons.
<input type="checkbox"/>	Metatag: Google Custom Search Engine (CSE) Provides support for meta tags used for Google Custom Search Engine.
<input type="checkbox"/>	Metatag: Google Plus Provides support for Google's Plus meta tags. Deprecated, will be removed in Metatag 2.0.0.
<input type="checkbox"/>	Metatag: Hreflang Provides support for the hreflang meta tag with some extra logic to simplify it.
<input type="checkbox"/>	Metatag: Mobile & UI Adjustments Provides support for meta tags used to control the mobile browser experience.
<input type="checkbox"/>	Metatag: Open Graph Provides support for Open Graph Protocol meta tags.
<input type="checkbox"/>	Metatag: Open Graph Products Provides additional Open Graph Protocol meta tags for describing products.
<input type="checkbox"/>	Metatag: Page Manager Provides metatag support for Page Manager variants.
<input type="checkbox"/>	Metatag: Pinterest Provides support for Pinterest's custom meta tags.
<input type="checkbox"/>	Metatag: Twitter Cards Provides support for Twitter's Card meta tags.
<input type="checkbox"/>	Metatag: Verification Verifies ownership of a site for search engines and other services.
<input type="checkbox"/>	Metatag: Views Provides views integration for metatags.

[Install](#)



Getting Started



This module can be downloaded with Composer.

- `composer require drupal/metatag`

Once downloaded, go to Extend Page and install the Metatag module, or use the drush command to enable it.

- `drush en metatag`

Along with meta tags, you can also install meta tags extensions such as an open graph and Twitter cards.

Note: The Meta tags module requires users to install tokens modules and their dependencies.



Filter

Metatag

^ SEO



Metatag

▼ Manage meta tags for all entities.

Install





Default Configuration Page



Once the Metatag module has been installed, navigate to `/admin/config/search/metatag` to get the default configuration screen.

From here, configure the meta tags across the entire website globally, just for the front page, or on a per-content type basis.

- Global: Configure the default meta tags across the website
- Front Page: Configure the meta tags specifically for the front page
- Content: Configure the meta tags for all content types
- Taxonomy: For taxonomy terms

Home Manage Shortcuts admin Admin Toolbar quick search Devel

Content Structure Appearance Extend Configuration People Reports Help

Metatag defaults Settings

Configure global meta tag default values below. Meta tags may be left as the default.
Meta tag patterns are passed down from one level to the next unless they are overridden. To view a summary of the individual meta tags and the pattern for a specific configuration, click on its name below.
If the top-level configuration is not specific enough, additional default meta tag configurations can be added for a specific entity type or entity bundle, e.g. for a specific content type.
Meta tags can be further refined on a per-entity basis, e.g. for individual nodes, by adding the "Metatag" field to that entity type through its normal field settings pages.

+ Add default meta tags

Type	Status	Operations
Global	Active	Edit
Front page	Active	Edit
403 access denied	Active	Edit
404 page not found	Active	Edit
Content	Active	Edit
Taxonomy term	Active	Edit
User	Active	Edit





Configure The Meta Tags Globally

- Configure the default meta tags for all web pages in the global config section. Whatever is defined here will be applied to all pages unless it is overridden by configuring in their respective configuration sections.
- There are two types of meta tags available by default:
 - Basic: These are the mandatory tags in Drupal that improve the overall Drupal SEO ranking.
 - Examples: Page Title, Description, Abstract, and Keyword.
 - Advance: These are optional tags used to declare the structured metadata for websites.
 - Examples: Geographical, Canonical URL, Robots, and Image.
- Let's see some examples of how this can be configured.



Configure Meta Tags For Content Type



- Add meta tags to show meta tag configuration options for specific content types.
- After adding the meta tags for the content type, it inherits them from Global and Content settings.
- I will now demonstrate how you can add default meta tags to your content type.



Manage Meta Tags For Node & View Pages

- The Metatag module provides a metatag field type.
- In the previous slides, some default meta tags were set for all nodes of different content types.
- However, users can override it on a per-node basis. To do this, add an extra field to the Content-Type of field type “Meta Tags.”
- A new tab will appear where one can add the meta tag values.



Add Meta Tags To View Pages

- The Meta Tag module provides a submodule called Meta Tag Views, allowing users to override meta tags for View pages.
- A new Setting for the View Page is available on the View edit page.
- We will demonstrate how you can override the meta tags for this View page.





Generate Twitter Cards Meta Tags

- When someone shares a link from a website, Twitter can use Twitter Cards to attach rich photos, videos, and media to Tweets.
- By configuring using the Metatag Module and the Twitter Cards submodule, users can see a “Card” added below the tweet containing neatly formatted information from the website.
- Enable the “Metatag: Twitter Card” submodule to use this functionality.
- Now I will demonstrate the “Summary Card with Large Image” Twitter cards.

Various Twitter Cards Available

- Summary Card: Displays Title, description, and thumbnail.
- Summary Card With Large Image: Similar to Summary Card but with a larger image.
- App Card: A Card with a direct download to a mobile app. Use this Card to drive app downloads.
- Player Card: Displays video, audio, or media.



Open Graph Meta Tags For Facebook

- Just like Twitter cards, a similar configuration can also be set for Facebook Meta tags.
- Enable the “Metatag: Open Graph” submodule to use this functionality.
- Now I will demonstrate this with examples.



A Few Major Tags

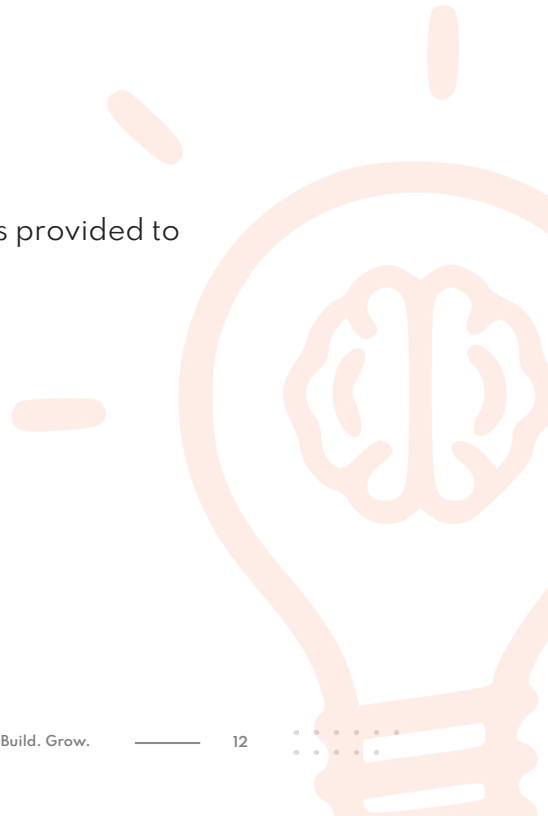
- og:url - URL of the page.
- og:title - Title of the page.
- og:image - Image related to content.
- og:description - Short description to justify the content page.
- og:site_name - Name of the websites.
- og:updated_time - Timestamp of the content update.





Customise Meta Tags Values Based On Conditions

- Users might sometimes want to disable all the meta tags for a particular route or path.
- This kind of customization can be achieved using the Meta tags hooks.
- Meta tags can also be obtained for an entity.
- For developers needing to access the rendered meta tags for a given entity, a function is provided to make this easy.
 - `$metatags = metatag_generate_entity_metatags($entity);`





Meta Tag Hook

```
/**
 * Alter the meta tags for pages that are not of content entities.
 *
 * @param array $metatags
 *   The special meta tags to be added to the page.
 * @param array $context
 *   The context for the current meta tags being generated. Will contain the
 *   following:
 *   'entity' - The entity being processed; passed by reference.
 */
0 references
function hook_metatags_alter(array &$metatags, array &$context) {
  // Exclude meta tags on frontpage.
  if (\Drupal::service('path.matcher')->isFrontPage()) {
    $metatags = NULL;
  }
  // Exclude meta tags on 'player' route.
  if (\Drupal::routeMatch()->getRouteName() === 'player') {
    $metatags = NULL;
  }
  // Setting metatag for custom form page.
  // Getting Current page path.
  $current_path = \Drupal::service('path.current')->getPath();
  if ($current_path == '/player/info') {
    $img_url = '/themes/custom/player/images/metatag-image/player_info.jpg';
    // Set the og:image meta tags for OpenGraph.
    $metatags['og_image'] = $img_url;
    $metatags['og_image_url'] = $img_url;
    // Set the meta tags for Twitter Cards.
    $metatags['twitter_cards_type'] = 'summary_large_image';
    $metatags['twitter_cards_image'] = $img_url;
  }
}
```





Q&A





Thank You