GN

Get meta tags programmatically

JAN 27, 2023 DRUPAL

To access the meta tags field data created by the Metatag module you can use the Metatag Manager service.

```
$metatag_manager = \Drupal::service('metatag.manager');
```

First, get the tags from the entity (usually from a node):

```
$tags = $metatag_manager->tagsFromEntityWithDefaults($node);
```

and then generate the meta tag values:

```
$metatags = $metatag_manager->generateTokenValues($tags, $node);
```

Using Entity Type Manager to get a list of fields

JAN 5, 2023 DRUPAL

Get all entity reference fields that target media entities

To get a list of all entity reference fields that target media entities you can use the Entity type manager and the *loadByProperties()* method.

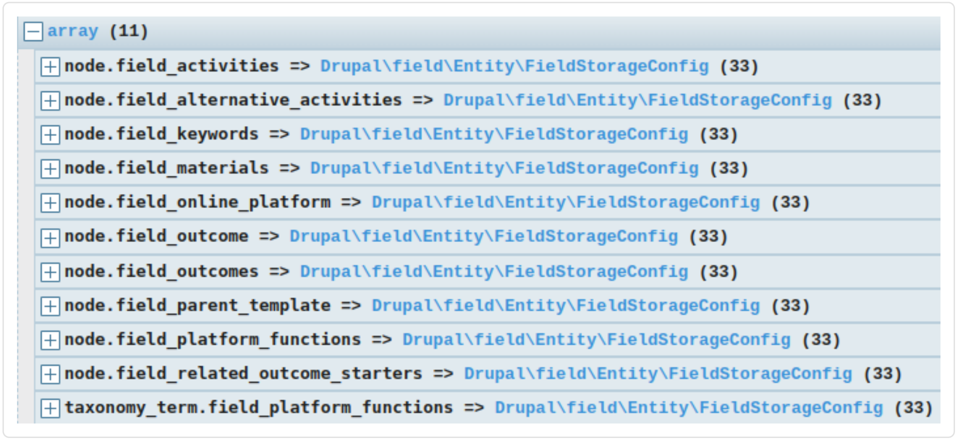
```
$fields = \Drupal::entityTypeManager()->getStorage('field_storage_config')->loadByProperties([
  'type' => 'entity_reference',
  'settings' => [
    'target_type' => 'media',
  ],
]);
```

Get all dynamic entity reference fields

In the same way, you can get a list of dynamic entity reference fields.

```
$fields = \Drupal::entityTypeManager()->getStorage('field_storage_config')->loadByProperties([
  'type' => 'dynamic_entity_reference',
]);
```

The *$fields* variable will contain an array of FieldStorageConfig instances for each field that satisfies the condition.


```
array (11)
  node.field_activities => Drupal\field\Entity\FieldStorageConfig (33)
  node.field_alternative_activities => Drupal\field\Entity\FieldStorageConfig (33)
  node.field_keywords => Drupal\field\Entity\FieldStorageConfig (33)
  node.field_materials => Drupal\field\Entity\FieldStorageConfig (33)
  node.field_online_platform => Drupal\field\Entity\FieldStorageConfig (33)
  node.field_outcome => Drupal\field\Entity\FieldStorageConfig (33)
  node.field_outcomes => Drupal\field\Entity\FieldStorageConfig (33)
  node.field_parent_template => Drupal\field\Entity\FieldStorageConfig (33)
  node.field_platform_functions => Drupal\field\Entity\FieldStorageConfig (33)
  node.field_related_outcome_starters => Drupal\field\Entity\FieldStorageConfig (33)
  taxonomy_term.field_platform_functions => Drupal\field\Entity\FieldStorageConfig (33)
```

This won't return fields that are defined as BaseFieldDefinition in your code.

How to zip files in Drupal

DEC 20, 2022 DRUPAL

Creating a zip archive in Drupal is easy. First, make sure that your PHP is compiled with ZIP support, then prepare the directory where you want to create a zip file, and finally create the zip file.

In the following example, I'm creating a zip archive by adding the file uploaded in the *field_file* field.

```
use Drupal\node\NodeInterface;
use Drupal\Core\File\FileSystemInterface;

function MY_MODULE_create_zip_file(NodeInterface $node) {
  // Check if ZipArchive class exists.
  if (!class_exists('ZipArchive')) {
    \Drupal::logger('MY_MODULE')->warning('Could not compress file, PHP is compiled without zip support.');
  }

  // Prepare destination directory.
  $directory = 'public://node-zip-files/' . $node->bundle();
  $file_system = \Drupal::service('file_system');
  $file_system->prepareDirectory($directory, FileSystemInterface:: CREATE_DIRECTORY | FileSystemInterface::MODIFY_PERMISSIONS);

  // Open zip archive.
  $zip = new \ZipArchive();
  $zip_filename = $file_system->realpath($directory . '/' . $node->id() . '.zip');
  $result = $zip->open($zip_filename, constant('ZipArchive::CREATE'));
  if (!$result) {
    \Drupal::logger('MY_MODULE')->warning('Zip archive could not be created. Error: ' . $result);
  }

  // Add file uploaded to the provided node to the zip acrhive.
  $file = $node->get('field_file')->entity;
  $filepath = $file_system->realpath($file->getFileUri());
  $result = $zip->addFile($filepath, basename($file->getFileUri()));
  if (!$result) {
    \Drupal::logger('MY_MODULE')->warning('File could not be added to zip archive.');
  }

  // Close zip archive.
  $result = $zip->close();
  if (!$result) {
    \Drupal::logger('MY_MODULE')->warning('Zip archive could not be closed.');
  }
}
```

This function can be used like this:

```
$node = \Drupal::entityTypeManager()->getStorage('node')->load(SOME_NODE_ID);
MY_MODULE_create_zip_file($node);
```

The zip file will be created in the following directory: *public://node-zip-files/NODE_BUNDLE/SOME_NODE_ID.zip*

How to alter a route - Drupal 9

DEC 5, 2022 DRUPAL

Sometimes you have to alter a route that is defined by a contrib module or Drupal core. Don't hack the module and change the **.routing.yml* file. That is a very bad idea because the next time you update the module those changes will be gone.

The best way to alter a route in Drupal 9 is to use a Route subscriber. In the following example, I had to change the allowed auth mechanism for the JWT auth issuer route.

MY_MODULE.services.yml

```
services:
  MY_MODULE.route_subscriber:
    class: Drupal\MY_MODULE\Routing\RouteSubscriber
    tags:
      - { name: event_subscriber }
```

src/Routing/RouteSubscriber.php

```
<?php

namespace Drupal\MY_MODULE\Routing;

use Drupal\Core\Routing\RouteSubscriberBase;
use Symfony\Component\Routing\RouteCollection;

class RouteSubscriber extends RouteSubscriberBase {

  /**
   * {@inheritdoc}
   */
  protected function alterRoutes(RouteCollection $collection) {
    if ($route = $collection->get('jwt_auth_issuer.jwt_auth_issuer_controller_generateToken')) {
      $route->setOption('_auth', ['basic_auth', 'email_basic_auth']);
    }
  }

}
```

Obviously, the $route object has a bunch of different useful methods like *$route->setDefault()* and *$route->setRequirement()* , so you can change anything you want.

Fix duplicate file names

NOV 28, 2022 DRUPAL

Drupal won't allow you to have two files with the same filename in a directory. Depending on how you put the file in the directory, Drupal will either rename it or replace the existing file. But it will allow you to have two or more files with the same filename inside different directories. To find and rename all such files you can use the following code snippet:

```
function MY_MODULE_fix_duplicate_filenames() {
  $connection = \Drupal::database();
  $query = $connection->query("SELECT filename FROM file_managed GROUP BY filename having count(*) >1;");
  $items = $query->fetchAll();

  /** @var \Drupal\file\FileStorageInterface $file_storage */
  $file_storage = \Drupal::entityTypeManager()->getStorage('file');
  /** @var \Drupal\file\FileRepositoryInterface $file_repository */
  $file_repository = \Drupal::service('file.repository');

  foreach ($items as $item) {
    $files = $file_storage->loadByProperties(['filename' => $item->filename]);

    /** @var \Drupal\file\FileInterface $file */
    foreach ($files as $file) {
      $pathinfo = pathinfo($file->getFilename());
      if (!empty($pathinfo['filename'])) {
        if (file_exists($file->getFileUri())) {
          $filename_suffix = time() . '-' . random_int(100000, 1000000);
          $new_destination = str_replace($pathinfo['filename'], $pathinfo['filename'] . '-' . $filename_suffix, $file->getFileUri());
          $file_repository->move($file, $new_destination);
        }
      }
    }
  }
}
```

As you can see, this will append a suffix constructed from a timestamp and a random integer value.

Enable Twig debug via Devel PHP

AUG 6, 2022 DRUPAL

Enabling Twig debug info via Devel PHP is not something I recommend you to do but in some rare cases, you might have to do it. Here's how to do it:

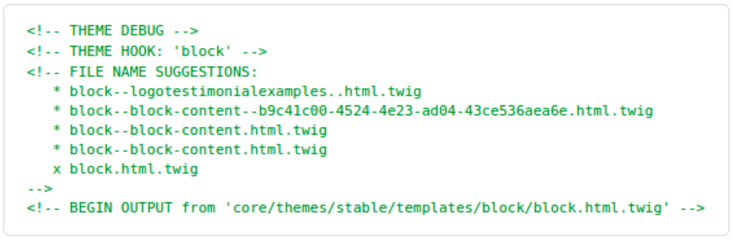
```
use Symfony\Component\Yaml\Yaml;

$yaml = Yaml::parseFile('sites/default/default.services.yml');
$yaml['parameters']['twig.config']['debug'] = TRUE;
$updated_yaml = Yaml::dump($yaml);
file_put_contents('sites/default/services.yml', $updated_yaml);
```

Nothing fancy here. Just loading the default services file, changing one variable to TRUE and then saving the content to services.yml file.

Template suggestions for Drupal 9 block types

APR 10, 2022 DRUPAL

By default Drupal 9 doesn't have a template suggestion for block types. If you enable theme debug and inspect a block you'll see something like this:



```
<!-- THEME DEBUG -->
<!-- THEME HOOK: 'block' -->
<!-- FILE NAME SUGGESTIONS:
   * block--logotestimonialexamples..html.twig
   * block--block-content--b9c41c00-4524-4e23-ad04-43ce536aea6e.html.twig
   * block--block-content.html.twig
   * block--block-content.html.twig
   x block.html.twig
-->
<!-- BEGIN OUTPUT from 'core/themes/stable/templates/block/block.html.twig' -->
```

To fix this and add a suggestion for all block types you have to implement the *hook_theme_suggestions_hook()* in your *.theme file:

```
/**
 * Implementation of hook_theme_suggestions_hook().
 */
function MY_THEME_theme_suggestions_block_alter(&$suggestions, $variables) {
  if (isset($variables['elements']['content']['#block_content'])) {
    $bundle = $variables['elements']['content']['#block_content']->bundle();
```

```
      array_splice($suggestions, 1, 0, 'block__' . $bundle);
    }
  }
```

If you take a look at theme debug info now you'll see something like this:

```
<!-- THEME DEBUG -->
<!-- THEME HOOK: 'block' -->
<!-- FILE NAME SUGGESTIONS:
   * block--logotestimonialexamples..html.twig
   * block--block-content--b9c41c00-4524-4e23-ad04-43ce536aea6e.html.twig
   * block--block-content.html.twig
   x block--image-block.html.twig
   * block--block-content.html.twig
   * block.html.twig
-->
<!-- BEGIN OUTPUT from 'themes/custom/testimonials_theme/templates/block/block--image-block.html.twig' -->
```

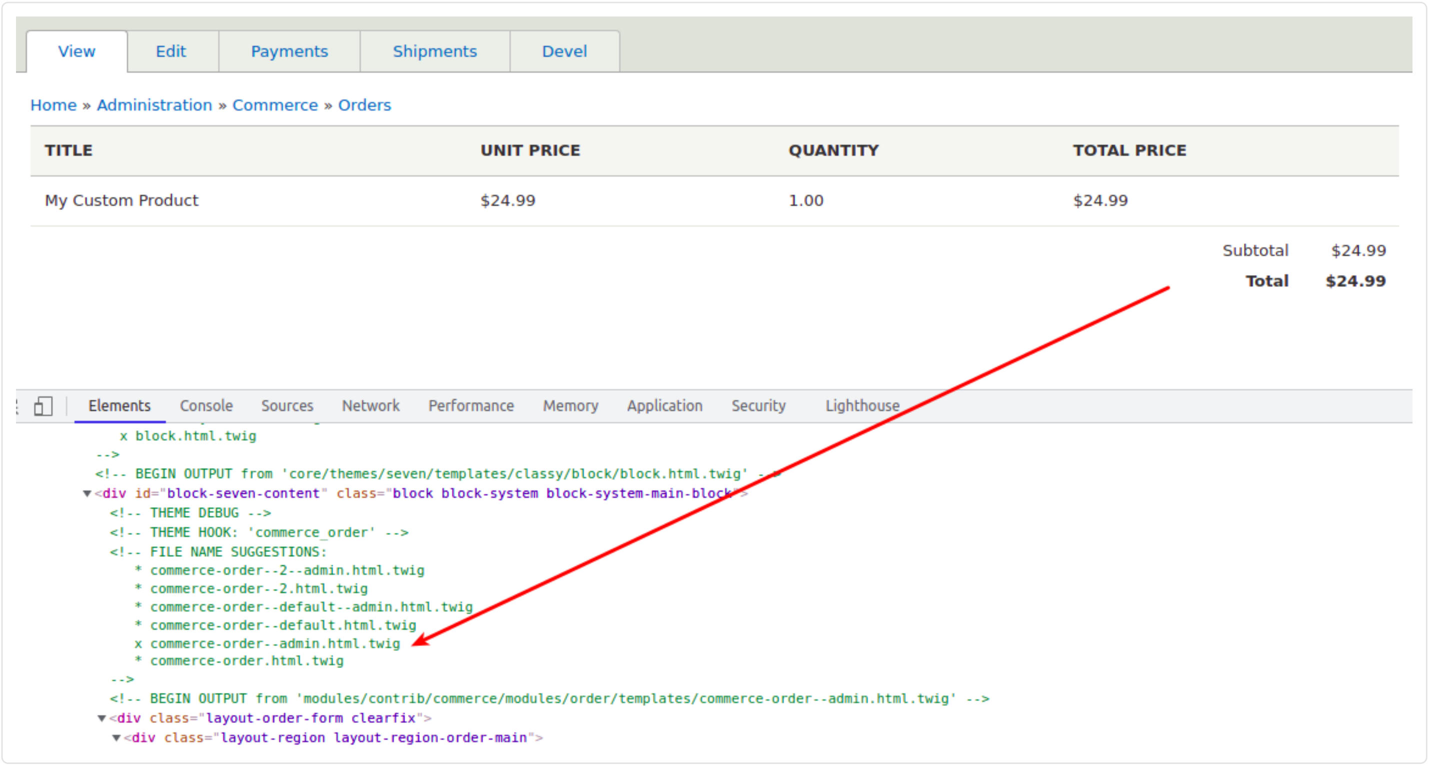
As you can see we now have a theme suggestion for the Image block type. Just by implementing one hook, you can now create a different Twig template for each block type.

Override admin template defined in a contrib module

DEC 15, 2021 DRUPAL

Let's say that you want to override the template for the order that is defined in the Commerce Order module. If you don't have a custom Admin theme then you have to override it in a custom module.

You can't do this in the custom user-facing theme, because that theme is not used for admin pages. So, you have to find out the template name by enabling theme debug and inspecting the HTML code.



TITLE	UNIT PRICE	QUANTITY	TOTAL PRICE
My Custom Product	$24.99	1.00	$24.99

Subtotal	$24.99
Total	**$24.99**

```
      x block.html.twig
    -->
    <!-- BEGIN OUTPUT from 'core/themes/seven/templates/classy/block/block.html.twig' -->
  ▼<div id="block-seven-content" class="block block-system block-system-main-block">
      <!-- THEME DEBUG -->
      <!-- THEME HOOK: 'commerce_order' -->
      <!-- FILE NAME SUGGESTIONS:
         * commerce-order--2--admin.html.twig
         * commerce-order--2.html.twig
         * commerce-order--default--admin.html.twig
         * commerce-order--default.html.twig
         x commerce-order--admin.html.twig
         * commerce-order.html.twig
      -->
      <!-- BEGIN OUTPUT from 'modules/contrib/commerce/modules/order/templates/commerce-order--admin.html.twig' -->
    ▼<div class="layout-order-form clearfix">
      ▼<div class="layout-region layout-region-order-main">
```

Now you can implement the *hook_theme()* hook like this:

```
/**
 * Implements hook_theme().
 */
function MY_MODULE_theme($existing, $type, $theme, $path) {
  return [
    'commerce_order__admin' => [
      'template' => 'commerce-order--admin-custom',
      ],
    ];
  }
```

And the last step is to create a new Twig template in your custom module: *MY_MODULE/templates/commerce-order--admin-custom.html.twig*

Altering the username that is displayed for a user

DEC 6, 2021 DRUPAL

Thanks to the *hook_user_format_name_alter()* hook altering the username that is displayed for a user is really easy. Let's say that you have the First and Last name fields on the user entity. You can do something like this to alter the username:

```
use Drupal\Core\Session\AccountInterface;
use Drupal\user\Entity\User;

/**
 * Implements hook_user_format_name_alter().
 */
function MY_MODULE_user_format_name_alter(&$name, AccountInterface $account) {
  $full_name = [];

  $account = User::load($account->id());
  $first_name = $account->get('field_first_name')->value;
  $last_name = $account->get('field_last_name')->value;

  if (!empty($first_name)) {
    $full_name[] = $first_name;
  }

  if (!empty($last_name)) {
    $full_name[] = $last_name;
  }

  if (!empty($full_name)) {
    $name = implode(' ', $full_name);
  }
}
```

If you now try to get the username by using this *$account->getDisplayName()* you will see the new username that consists of the first and last name.

How to alter Read more link for a node

DEC 1, 2021 DRUPAL

If you want to alter the title of the *Read more* link or perhaps add some CSS classes, you can do that by using the *hook_node_links_alter()* hook.

```
use Drupal\node\NodeInterface;

function MY_MODULE_node_links_alter(array &$links, NodeInterface $entity, array &$context) {
  if (!empty($links['node']['#links']['node-readmore']['title'])) {
    $links['node']['#links']['node-readmore']['title'] = t('View Details');
    $links['node']['#links']['node-readmore']['attributes']['class'][] = 'view-details-link';
  }
}
```

You can do this just for some content types, by checking the bundle value: *if ($entity->bundle() == 'article') {}*

1 2 3 4 Next › Last »