

[Blog](#) [Guides](#) [Snippets](#) [Contact](#) [About](#)

Create media programmatically in Drupal

📅 OCT 1, 2022 💎 DRUPAL

Using Media in Drupal (versions 8, 9, and 10) is a much better option than using old-school image fields. You can add all types of fields to Media to enrich your images, videos, and other media types with additional information, and you can also reuse Media. All of that cannot be easily done if you are using regular image fields.

Just like Content types, you can have multiple different types of Media. By default when you enable the Media module you get the following types:

Media types ☆	
+ Add media type	
Name	Description
Audio	A locally hosted audio file.
Document	An uploaded file or document, such as a PDF.
Image	Use local images for reusable media.
Remote video	A remotely hosted video from YouTube or Vimeo.
Video	A locally hosted video file.

If you analyze the Image media type, you will see that by default there is only one field. That field is `field_media_image` which is of the Image type. Let's add some additional fields to enrich our images and provide more useful information to our users. In the following screenshot, you can see that I added the Author, Date, and Location fields.

Manage fields ☆

[Edit](#)
[Manage fields](#)
[Manage form display](#)
[Manage display](#)
[Manage permissions](#)
[Devel](#)

+ Add field

Label	Machine name	Field type
Author	field_author	Text (plain)
Date	field_date	Date
Image	field_media_image	Image
Location	field_location	Text (plain)

Before we move on with programmatically creating Media entities, you can also check my article on [how to convert existing image fields to media images](#). It's fairly easy to write an update hook that will do the conversion automatically.

To programmatically save a media entity of the image type you can do something like this:

```
use Drupal\media\Entity\Media;

$image_id = 1;

$image_media = Media::create([
  'name' => 'My media name',
  'bundle' => 'image',
  'uid' => 1,
  'langcode' => 'en',
  'status' => 0,
  'field_media_image' => [
    'target_id' => $image_id,
    'alt' => t('My media alt attribute'),
    'title' => t('My media title attribute'),
  ],
  'field_author' => 'Goran Nikolovski',
  'field_date' => '2025-12-31T23:59:59',
  'field_location' => 'Subotica',
]);
$image_media->save();
```

The example above assumes that you have the image file ID. This could be the case when you are converting image fields to media. But you can also [create and save an image file in Drupal programmatically](#):

```
use Drupal\media\Entity\Media;
use Drupal\Core\File\FileSystemInterface;

$image_data = file_get_contents('https://gorannikolovski.com/themes/custom/gn2021/images/goran-nikolovski-signature.png');
$file_repository = \Drupal::service('file.repository');
$image = $file_repository->writeData($image_data, "public://goran-nikolovski-signature.png", FileSystemInterface::EXISTS_REPLACE);

$image_media = Media::create([
  'name' => 'My media name',
  'bundle' => 'image',
  'uid' => 1,
  'langcode' => 'en',
  'status' => 0,
  'field_media_image' => [
```

```

    'target_id' => $image->id(),
    'alt' => t('My media alt attribute'),
    'title' => t('My media title attribute'),
  ],
  'field_author' => 'Goran Nikolovski',
  'field_date' => '2025-12-31T23:59:59',
  'field_location' => 'Subotica',
]);
$image_media->save();

```

As you can see, creating a media entity is very similar to creating other entities in Drupal. If you take a closer look, you will see that we don't have *title* and *type* fields like nodes have, but instead, we have *name* and *bundle* fields.

As always you don't have to hardcode the *uid* and *langcode* values. You can set the current user ID and the current language code:

```

$current_user_id = \Drupal::currentUser()->id();

$language_code = \Drupal::languageManager()->getCurrentLanguage()->getId();

```

How to create a Media video youtube entity programmatically?

To do this you have to use the *Remote video* media type.

```

use Drupal\media\Entity\Media;

$video_media = Media::create([
  'bundle' => 'remote_video',
  'uid' => 1,
  'name' => 'DrupalCon Prague 2022 Driesnote',
  'field_media_oembed_video' => [
    'value' => 'https://www.youtube.com/watch?v=Hmit4ET-l3Q',
  ],
]);
$video_media->save();

```

Finally, you can attach the created media entity to some other entity. In the following example, we are attaching it to the Basic page content type:

```

use \Drupal\node\Entity\Node;

$node = Node::create([
  'type' => 'page',
  'title' => 'My media parent node',
  'field_youtube' => [
    'target_id' => $video_media->id(),
  ],
]);
$node->save();

```

The media module has been added to Drupal core back in version 8.4 (5 years ago) so it's about time we all start using it.

That's pretty much it. Now go on and create some media entities in your code.

[#drupal](#) [#media](#) [#programmatically](#)

Further reading

- [Create file programmatically in Drupal](#)
- [How to convert existing image fields into Media images](#)
- [Programmatically create a menu link for a node](#)