


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

## How to add a button to Drupal 8/9 Modal

NOV 29, 2021 DRUPAL

Creating a modal with a title and some content in Ajax callback is easy:

```
use Drupal\Core\Form\FormStateInterface;
use Drupal\Core\Ajax\AjaxResponse;
use Drupal\Core\Ajax\OpenModalDialogCommand;

public function ajaxCallback(array $form, FormStateInterface $form_state) {
    $response = new AjaxResponse();
    $response->addCommand(new OpenModalDialogCommand($this->t('Title!'), $this->t('Some content.')));
    return $response;
}
```

If you also want to display a button in the modal, just provide the third parameter like this:

```
$options = [
    'buttons' => [
        'button1' => [
            'text' => $this->t('Close'),
            'id' => 'close-button',
            'onclick' => "jQuery('.ui-icon-closethick').click()",
        ],
    ],
];

...
$response->addCommand(new OpenModalDialogCommand($this->t('Title!'), $this->t('Some content.'), $options));
...
```

As you can see, by clicking on this button the modal will be closed.

## How to programmatically check Generate automatic URL alias

NOV 9, 2021 DRUPAL

If you want to programmatically check the Pathauto's *Generate automatic URL alias* checkbox you can do something like this:

```
use Drupal\pathauto\PathautoState;

$entity_type = 'node';
$entity_storage = \Drupal::entityTypeManager()->getStorage($entity_type);
$nodes = $entity_storage->loadMultiple();

foreach($nodes as $node) {
    $node->path->pathauto = PathautoState::CREATE;
    $node->save();
}
```

For terms just change the entity type to this:

```
$entity_type = 'taxonomy_term';
```

To uncheck the field instead of CREATE use SKIP constant:

```
$node->path->pathauto = PathautoState::SKIP;
```

## How to programmatically render entity form

SEP 14, 2021 DRUPAL

Programmatically rendering entity form in Drupal 8 and 9 is easy, provided that you want to render it using the default form mode.

```
// Load existing node
$node = \Drupal\node\Entity\Node::load(1);
// or create a new node
$node = \Drupal::entityTypeManager()->getStorage('node')->create(['type' => 'article']);

$form = \Drupal::service('entity.form_builder')->getForm($node);
```

If you want to render it using any other form mode like this for example:

```
$node = \Drupal\node\Entity\Node::load(1);
$form = \Drupal::service('entity.form_builder')->getForm($node, 'compact');
```

you will have to alter the entity definition and add your own form handler. Otherwise, you'll get the following error:

```
Drupal\Component\Plugin\Exception\InvalidPluginDefinitionException: The "node" entity type did not specify a "compact" form class.
```

To add your own form class you can use the `hook_entity_type_alter` hook like this:

```
/**
 * Implements hook_entity_type_alter().
 *
 * Alters the entity definition and adds our own form handlers.
 */
function MY_MODULE_entity_type_alter(array &$entity_types) {
  $form_modes = \Drupal::service('entity_display.repository')
    ->getAllFormModes();

  foreach ($form_modes as $entity_type => $display_modes) {
    if ($entity_type !== 'node') {
      continue;
    }

    $type = $entity_types[$entity_type];
    foreach ($display_modes as $machine_name => $form_display) {
      if (isset($type->getHandlerClasses()['form']['default'])) {
        $default_handler_class = $type->getHandlerClasses()['form']['default'];
        $type->setFormClass($machine_name, $default_handler_class);
      }
    }
  }
}
```

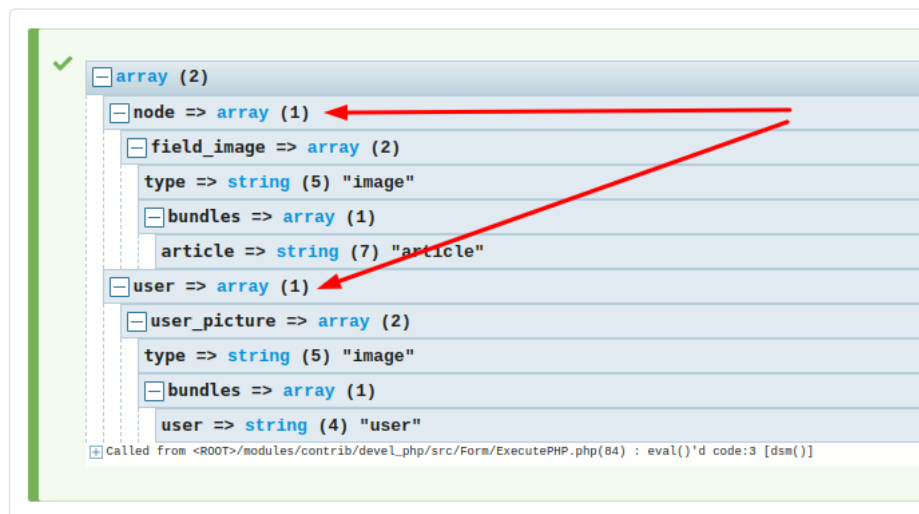
## How to get all field names of field type

JUL 16, 2021 DRUPAL

To get a list of all field names for the fields whose field type is for example `image` you can use the `entity_field.manager` service:

```
$field_map = \Drupal::service('entity_field.manager')->getFieldMapByFieldType('image');
```

The `field_map` array will contain a list of fields by entity type.



## How to programmatically change order status

📅 JUL 9, 2021 🐾 DRUPAL

Changing the order state in Drupal Commerce 2.x like this:

```
$order->set('state', 'canceled');
$order->save();
```

is not a good idea because you are not dispatching transition events. And some modules depend on those events. The proper way to change the order state is to apply state transition like this:

```
$order = \Drupal::entityTypeManager()
  ->getStorage('commerce_order')
  ->load(1);
$order->getState()->applyTransitionById('cancel');
$order->save();
```

You can find the list of transition IDs in the [workflows file](#).

## How to programmatically add a language

📅 MAY 25, 2021 🐾 DRUPAL

Sometimes you have to add a new language in Drupal 8/9 programmatically. For example, maybe you need to write an update hook that will add a new language. This is how you can do it:

```
use Drupal\language\Entity\ConfigurableLanguage;

$language = ConfigurableLanguage::createFromLangcode('sr');
$language->save();
```

If you don't know the langcode, you can find it by inspecting the select box on the `/admin/config/regional/language/add` page.

## How to set price field value programmatically in Drupal Commerce

📅 MAY 6, 2021 🐾 DRUPAL

Price in Drupal Commerce 2.x is not a number but a class. This class has a price number and currency code properties, and both are strings. So, to set the product price programmatically you have to first instantiate the Price class:

```
$price = new \Drupal\commerce_price\Price('9.99', 'EUR');
```

and then use created Price object to set the product price:

```
$product_variation = \Drupal\commerce_product\Entity\ProductVariation::load(1);
$product_variation->set('price', $price);
$product_variation->save();
```

To get the price number and currency code you can use appropriate getters methods:

```
$price_number = $price->getNumber();
$currency_code = $price->getCurrencyCode();
```

You can also create a Price object by using the static method:

```
$price = Price::fromArray(['number' => '9.99', 'currency_code' => 'EUR']);
```

## How to customize the Drupal Commerce order receipt email subject

📅 APR 28, 2021 🐘 DRUPAL

At the moment you can't configure the order receipt email subject. The subject is hardcoded in the [OrderReceiptMail](#) service.

There is an [issue to fix this](#), but until that happens the only way to alter the email subject is to use the `hook_mail_alter()`.

```
/**
 * Implements hook_mail_alter().
 */
function MY_MODULE_mail_alter(&$message) {
  if ($message['key'] == 'order_receipt' && !empty($message['params']['order'])) {
    /** @var \Drupal\commerce_order\Entity\OrderInterface $order */
    $order = $message['params']['order'];

    $message['from'] = 'noreply@example.com';
    $message['subject'] = t('Receipt for Order ID: @number', [
      '@number' => $order->getOrderNumber(),
    ]);
  }
}
```

## How to generate masquerade link programmatically

📅 MAR 1, 2021 🐘 DRUPAL

The [Masquerade module](#) allows website admins to switch users and visit the website as that user. This is helpful for developers and site builders when trying to find out what a client might see when logged into the website. To create the unmasquerade programmatically use this:

```
if (
  \Drupal::moduleHandler()->moduleExists('masquerade') &&
  \Drupal::service('masquerade')->isMasquerading()
){
  $links['unmasquerade'] = [
    '#type' => 'link',
    '#title' => t('Unmasquerade'),
    '#url' => Url::fromRoute('masquerade.unmasquerade'),
    '#weight' => 1500,
  ];
}
```

## Drupal 8/9 and Guzzle HTTP client library

📅 NOV 30, 2020 🐘 DRUPAL

Making a simple HTTP Get request in Drupal 8 or 9 is super easy thanks to the Guzzle library. This library has been added to Drupal core a long time ago, and it replaced the much less powerful `drupal_http_request()` function. Let's see how to get some data from the Star Wars API:

```
use GuzzleHttp\Client;
use GuzzleHttp\Exception\RequestException;

$client = new Client();
$people = [];

try {
    $response = $client->get('https://swapi.dev/api/people');
    $result = json_decode($response->getBody(), TRUE);
    foreach($result['results'] as $item) {
        $people[] = $item['name'];
    }
}
catch (RequestException $e) {
    // log exception
}
```

Guzzle can also be accessed through the `http_client` service and the helper method:

```
$client = \Drupal::httpClient();
```

Sending a POST request is also simple and easy.

```
public function sendOrder(OrderInterface $order) {
    try {
        $response = $this->httpClient->post($url, [
            'form_params' => $data,
        ]);
        $response_data = json_decode($response->getBody()->getContents(), TRUE);

        // do something with data
    }
    catch (RequestException $e) {
        // log exception
    }
}
```

If you want to log the exception don't use `$e->getMessage()` because the error message will be truncated. Instead, use this:

```
$e->getResponse()->getBody()->getContents();
```

You can also use Guzzle to send files. To send an image file you can do something like this:

```
try {
    $client = new Client(['headers' => ['X-Auth-Secret-Token' => 'my-secret-token']]);
    $options = [
        'multipart' => [
            [
                'Content-type' => 'multipart/form-data',
                'name' => 'image',
                'contents' => fopen($image->getUri(), 'r'),
                'filename' => basename($image->getUri()),
            ],
        ],
    ];
    $response = $client->post($api_base_url . 'api/v1/images', $options);
}
catch (\Exception $e) {
```

```
// log exception  
}
```

In this case, the **\$image** variable is an instance of *Drupal\file\Entity\File* class.

« First ‹ Previous 1 2 3 4 Next › Last »

Copyright © 2017-2023 by Goran Nikolovski. All rights reserved. [Top of page](#).

[Privacy Policy](#) | [Terms of Service](#) | [Article Tags](#) | [Search](#)

