

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

## Change status message text

OCT 22, 2020 DRUPAL

Let's say that a contrib module is showing a message like this: *Your message has been created*, and you want to change it to something like this: **Message created**. To do this you don't have to hack the contrib module, you can use the `hook_preprocess_HOOK()` to alter the message.

```
/**
 * Implements hook_preprocess_HOOK().
 */
function MY_MODULE_preprocess_status_messages(&$variables) {
  if (isset($variables['message_list']['status'])){
    $status_messages = $variables['message_list']['status'];
    foreach($status_messages as $delta => $message) {
      if (strpos((string) $message, 'Your message has been created.') !== FALSE) {
        $variables['message_list']['status'][$delta] = t('Message created.');
      }
    }
  }
}
```

## How to alter the page title?

OCT 22, 2020 DRUPAL

One way to alter the page title is to use `hook_preprocess_HOOK()` and then set the new page title depending on the current route name.

```
function MY_MODULE_preprocess_page_title(&$variables) {
  $route_name = \Drupal::routeMatch()->getRouteName();
  if ($route_name === 'entity.user.edit_form') {
    $variables['title'] = 'Edit My Account';
  }
}
```

This will only change the title in the Page title block. To actually change the `<title>` HTML tag you can use the following code snippet:

```
function MY_MODULE_preprocess_html(&$variables) {
  $route_name = \Drupal::routeMatch()->getRouteName();
  if ($route_name === 'entity.user.edit_form') {
```

```

$variables['head_title']['title'] = t('Edit My Account')
}
}

```

## Hide fields in Inline Entity Form

📅 OCT 13, 2020 💎 DRUPAL

If you want to hide a field inside the Inline Entity Form you have to use the *hook\_inline\_entity\_form\_entity\_form\_alter()* hook – standard form alter hook won't work. For example, to hide the status field you can do something like this:

```

/**
 * Implements hook_inline_entity_form_entity_form_alter().
 */
function MY_MODULE_inline_entity_form_entity_form_alter(&$entity_form, &$form_state) {
  if (in_array($entity_form['#entity_type'], ['YOUR_ENTITY_TYPE_1', 'YOUR_ENTITY_TYPE_2'])) {
    $entity_form['status']['#access'] = FALSE;
  }
}

```

## Disable Add to Cart button based on flag

📅 OCT 10, 2020 💎 DRUPAL

Hiding the Add to Cart button in Drupal Commerce 2.x is easy. Just alter the **commerce\_order\_item\_add\_to\_cart\_form** form, and set the disabled property of the button to true.

```

use Drupal\Core\Form\FormStateInterface;

function MY_MODULE_form_alter(&$form, FormStateInterface $form_state, $form_id) {
  if (strpos($form_id, 'commerce_order_item_add_to_cart_form') !== FALSE) {
    $product = $form_state->getFormObject()->getEntity()->getPurchasedEntity()->getProduct();

    $flag_service = \Drupal::service('flag');
    $flag = $flag_service->getFlagById('sold_out');
    $flagging = $flag_service->getFlagging($flag, $product);

    if ($flagging !== NULL ) {
      $form['actions']['submit']['#value'] = t('Out of stock');
      $form['actions']['submit']['#disabled'] = TRUE;
    }
  }
}

```

## Always show payment methods

📅 OCT 10, 2020 💎 DRUPAL

If you have only one payment method, then Drupal Commerce 2.x will hide it. We can change that behavior in the form alter hook.

```
use Drupal\Core\Form\FormStateInterface;

function MY_MODULE_form_alter(&$form, FormStateInterface $form_state, $form_id) {
  if ($form_id == 'commerce_checkout_flow_multistep_default') {
    if (isset($form['payment_information']['payment_method']['#access'])) {
      $form['payment_information']['payment_method']['#access'] = TRUE;
    }
  }
}
```

## Programatically add recaptcha to Drupal Commerce checkout form

SEP 10, 2020 DRUPAL

If you are using the UI you can add a Recaptcha only to the entire checkout form, and that usually is not what you want. Fortunately, adding a Recaptcha to a certain step is easy:

```
use Drupal\Core\Form\FormStateInterface;

function MY_MODULE_form_alter(&$form, FormStateInterface $form_state, $form_id) {
  if ($form_id == 'commerce_checkout_flow_multistep_default' && $form['#step_id'] == 'order_information') {
    $form['captcha'] = [
      '#type' => 'captcha',
      '#captcha_type' => 'recaptcha/reCAPTCHA',
    ];
  }
}
```

## Add a role on user create

SEP 5, 2020 DRUPAL

In some cases, you may want to add a role when a new user is created regardless of how the user is created (UI, GraphQL, Rest API, or something else). The `hook_ENTITY_TYPE_presave()` hook is perfect for this.

```
use Drupal\user\UserInterface;

function MY_MODULE_user_presave(UserInterface $user) {
  if ($user->isNew() && !$user->hasRole('frontend_app')) {
    $user->addRole('frontend_app');
  }
}
```

# Add a menu link to existing menu

SEP 5, 2020 DRUPAL

Adding a menu link to some existing menu programmatically (for example in an update hook) is something I have to do fairly often. Here's how you can add a new menu link to the main menu:

```
use Drupal\menu_link_content\Entity\MenuLinkContent;

function MY_MODULE_update_8001() {
  MenuLinkContent::create([
    'title' => 'About',
    'link' => ['uri' => 'internal:/about'],
    'menu_name' => 'main',
    'weight' => -50,
  ])->save();
}
```

And if you want to add a submenu link you first have to load the parent link and then create a submenu link:

```
use Drupal\menu_link_content\Entity\MenuLinkContent;

function MY_MODULE_update_8001() {
  $menu_link_parents = \Drupal::entityTypeManager()
    ->getStorage('menu_link_content')
    ->loadByProperties([
      'title' => 'Dashboard',
      'menu_name' => 'main',
    ]);

  $menu_link_parent = reset($menu_link_parents);

  if ($menu_link_parent) {
    MenuLinkContent::create([
      'title' => 'Add tasks',
      'link' => ['uri' => 'internal:/add-task'],
      'menu_name' => 'main',
      'weight' => -50,
      'parent' => $menu_link_parent->getPluginId(),
    ])->save();
  }
}
```

# Render a menu in a form

AUG 29, 2020 DRUPAL

Rendering a menu in a form requires a little bit more lines of code than rendering a View, but it is also very easy thanks to the *menu.link\_tree* service. In the following example, we are rendering the main menu.

```
use Drupal\Core\Menu\MenuTreeParameters;

$menu_parameters = new MenuTreeParameters();
$manipulators = [
  ['callable' => 'menu.default_tree_manipulators:checkNodeAccess'],
  ['callable' => 'menu.default_tree_manipulators:checkAccess'],
  ['callable' => 'menu.default_tree_manipulators:generateIndexAndSort'],
];
$tree = \Drupal::menuTree()->load('main', $menu_parameters);
$tree = \Drupal::menuTree()->transform($tree, $manipulators);

$form['content'] = \Drupal::menuTree()->build($tree);
```

## Render a View in a form

📅 AUG 25, 2020 💎 DRUPAL

Rendering Views in a form is super easy thanks to the *buildRenderable()* method, which builds the render array for the given display. To render a View, you just need to know the machine name of the View and display ID (usually page\_1 or block\_1). To render the Content View do this:

```
use Drupal\views\Views;

$view = Views::getView('content');
$form['content'] = $view->buildRenderable('page_1');
```

« 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](#)

