GN

Get the list of fields for an entity type

AUG 15, 2020 DRUPAL

To get the field definitions of any entity type use the following code. We first get all bundles, and then for each bundle we are getting the field definitions.

```
$bundles = \Drupal::service('entity_type.bundle.info')->getBundleInfo('node');

foreach ($bundles as $bundle => $data) {
  $fields = \Drupal::service('entity_field.manager')->getFieldDefinitions('node', $bundle);
}
```

Add CSS and JS to Admin pages

AUG 14, 2020 DRUPAL

Adding CSS or JS to admin pages is easy. Just create a new library, create CSS and JS files, and then attach the library in the *hook_page_attachments()* hook.

MY_MODULE.libraries.yml

```
admin_styling:
  version: VERSION
  css:
    theme:
      css/admin-styling.css: {}
  js:
    js/admin-script.js: {}
```

MY_MODULE.module

```
/**
 * Implements hook_page_attachments().
 */
function MY_MODULE_page_attachments(array &$attachments) {
  if (\Drupal::service('router.admin_context')->isAdminRoute()) {
    $attachments['#attached']['library'][] = 'MY_MODULE/admin_styling';
```

```
  }
}
```

Hide generator metatag

AUG 12, 2020 DRUPAL

If you want to hide the generator metatag that exposes information about your Drupal version then you can use the *hook_page_attachments_alter()* hook. This applies only for versions 8 and 9!

```
/**
 * Implements hook_page_attachments_alter().
 */
function MY_MODULE_page_attachments_alter(array &$attachments) {
  foreach ($attachments['#attached']['html_head'] as $key => $attachment) {
    if (isset($attachment[1]) && $attachment[1] == 'system_meta_generator') {
      unset($attachments['#attached']['html_head'][$key]);
    }
  }
}
```

Ajax form element callback

AUG 9, 2020 DRUPAL

To save the API token field without submitting the form, you can add a simple ajax callback function. In this example, we are saving the field value on the *keyup* event, but you can also use the *change* event or if you have an autocomplete field then you probably want to use the *autocompleteclose* event.

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

public function buildForm(array $form, FormStateInterface $form_state) {
  $form['api_token'] = [
    '#type' => 'textfield',
    '#title' => $this->t('API token'),
    '#ajax' => [
      'callback' => '::updateApiToken',
      'event' => 'keyup',
    ],
  ];

  return parent::buildForm($form, $form_state);
}
```

```
public function updateApiToken(array $form, FormStateInterface $form_state) {
  $response = new AjaxResponse();
  $this->config('MY_MODULE.settings')
    ->set('api_token', $form_state->getValue('api_token'))
    ->save();
  return $response;
}
```

Remove an item from entity reference field

AUG 2, 2020 DRUPAL

Let's say that you have a multivalue reference field named *field_tags* in a node type. To remove a specific item from that field you can do something like this:

```
$node_id = 152;
$tag_to_remove = 56;
$entity_storage = \Drupal::entityTypeManager()->getStorage('node');
$entity = $entity_storage->load($node_id);

foreach ($entity->get('field_tags') as $delta => $field_item) {
  if ($field_item->target_id == $tag_to_remove) {
    $entity->get('field_tags')->removeItem($delta);
    $entity->save();
    break;
  }
}
```

Redirect after login

JUL 31, 2020 DRUPAL

Adding a redirect to *hook_user_login()* is a common mistake. You should never add the redirect there because redirecting users in that hook would stop other *hook_user_login()* implementations from being invoked. To properly redirect users to some page after they log in, you should add a custom submit handler to the user login form.

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

/**
 * Implements hook_form_FORM_ID_alter().
 */
function MY_MODULE_form_user_login_form_alter(&$form, FormStateInterface $form_state, $form_id) {
  $form['#submit'][] = 'MY_MODULE_user_login_form_submit';
```

```
}

/**
 * Custom submit handler for the login form.
 */
function MY_MODULE_user_login_form_submit($form, FormStateInterface $form_state) {
  $url = Url::fromRoute('<front>');
  $form_state->setRedirectUrl($url);
}
```

« First ‹ Previous 1 2 3 4

Privacy Policy | Terms of Service | Article Tags | Search