Taxonomy

TABLE OF CONTENTS

- Lookup term by name
- Lookup term name using its tid
- Lookup term using its uuid
- Load terms from a term reference field
- Find terms referenced in a paragraph in a term reference field
- Get URL alias from a term ID
- Load all terms for a vocabulary
- Load all terms for a vocabulary and put them in a select (dropdown)
- Create taxonomy term programatically
- Find all nodes with a matching term
- Find nodes with a matching term using entityQuery

views 87

Lookup term by name

```
public function loadTermByName() {
  $vocabulary_id = 'event_category';
  $term_name = 'protest';
  $terms = \Drupal::entityTypeManager()
    ->getStorage('taxonomy_term')
    ->loadByProperties(['name' => $term_name]);

  if (empty($terms)) {
    $build['result'] = [
      '#type' => 'item',
      '#markup' => $this->t('No terms found'),
    ];
  }
  else {
    /** @var Term $term */
    foreach ($terms as $term) {
      $term_name = $term->getName();
      $build[$term_name] = [
        '#type' => 'item',
        '#markup' => $this->t('Term name: @term_name. Term id: @term_id', [
          '@term_name' => $term_name,
          '@term_id' => $term->id(),
        ]),
      ];
    }
  }

  return $build;
}

// Or the deprecated version.

$terms = taxonomy_term_load_multiple_by_name($term_name, 'opinion_categories');
if (empty($terms)) {
  $ra = [
    '#markup' => $this->t('Invalid term: @term', ['@term' => $category])
  ];
  return $ra;
}
// pop off the first one and grab it's term_id
$term = reset($terms);
$term_id = $term->get('tid')->value;
```

Lookup term name using its tid

```
use Drupal\taxonomy\Entity\Term;
$term = Term::load($term_id);

// OR

$term_name = \Drupal\taxonomy\Entity\Term::load($term_id)->label();

// Or

$term_name = \Drupal\taxonomy\Entity\Term::load($term_id)->get('name')->value;

// Or

$term = \Drupal::entityTypeManager()->getStorage('taxonomy_term')->load($term_id);

if (empty($term)) {
  $render_array = [
    '#markup' => $this->t('Invalid term id: @termid', ['@termid' => $term_id])
  ];
}
else {
  $term_name = $term->name;
  $render_array = [
    '#markup' => $this->t('Term name: @term_name', ['@term_name' => $term_name])
  ];

}
  return $render_array;
}
```

Lookup term using its uuid

Each taxonomy term has a UUID. See the taxonomy_term_data table uuid field. We can load a taxonomy term by it's uuid as shown below:

Here we load a taxonomy term, get it's name and it's tid.

```
public function loadByUUID() {
  $uuid = 'd4a7bbc5-3b1b-46a4-bea4-01255365999f';

  $storage = \Drupal::entityTypeManager()->getStorage('taxonomy_term');
  $term_loaded_by_uuid = $storage->loadByProperties(['uuid' => $uuid]);
  $term = reset($term_loaded_by_uuid);
  $str = "Term not found";
  if (!empty($term)) {
    $term_name = $term->getName();
    $term_id = $term->id();
    $str =  "Term uuid = $uuid<br>";
    $str .=  "Term name = $term_name<br>";
    $str .= "Term id = $term_id<br>";
  }

  $build['content'] = [
    '#type' => 'item',
    '#markup' => $str,
  ];

  return $build;
}
```

Load terms from a term reference field

Retrieve the values in the field_event_category and display their term name, term id and their uuid. The call to referencedEntities() returns an array of term objects, so no need to call load() on them separately.

```
public function loadTermRef() {
  $nid = 24;
  $event_node = Node::load($nid);
  if ($event_node) {
    $categories = $event_node->get('field_event_category')->referencedEntities();
  }
  $str = "Terms:<br>";

  /** @var \Drupal\Core\Entity\EntityInterface $category */
  foreach ($categories as $category) {
    $term_id = $category->id();
    $term_name = $category->getName();
    $uuid = $category->uuid();
    $str .= "Term name = $term_name<br>";
    $str .= "Term id = $term_id<br>";
    $str .= "Term uuid = $uuid<br>";
  }

  $build['content'] = [
    '#type' => 'item',
    '#markup' => $str,
  ];

  return $build;
}
```

Find terms referenced in a paragraph in a term reference field

Loop thru all the instances of a paragraph reference and grab the term in the paragraph.

use Drupal\taxonomy\Entity\Term;

```
foreach ($node->get('field_my_para')->referencedEntities() as $ent){
  $term = Term::load($ent->$field_in_paragraph->target_id);
  $name = $term->getName();
  print_r($name);
}
```

Get URL alias from a term ID

This will return something like:

```
URL Alias for tid 3 = https://d9book2.ddev.site/category/rally
```

```
public function getTaxonomyAlias() {
  $tid = 3;
  //return taxonomy alias
  $options = ['absolute' => TRUE];  //FALSE will return relative path.

  // Build a URL.
  /** @var \Drupal\Core\Url $url */
  $url = Url::fromRoute('entity.taxonomy_term.canonical', ['taxonomy_term' => $tid], $options);
  $path_string = $url->toString();
  $str = "URL Alias for tid $tid = $path_string";

  $build['content'] = [
    '#type' => 'item',
    '#markup' => $str,
  ];

  return $build;
}
```

Load all terms for a vocabulary

This code loads the terms into an array and displays them on screen. Note that you can't use id() or getName() on the objects returned from loadTree() as they are standard objects. If you load the actual term entities using Term::load(), then you can use entity functions like id() and getName(). It returns this:

Found 5 terms in vocabulary event_category
Term: Hunger strike term_id: 5
Term: Protest term_id: 4
Term: Rally term_id: 3
Term: Training term_id: 2
Term: Webinar term_id: 1

```
public function loadTerms() {
  $vocabulary_id = 'event_category';

  $terms = \Drupal::entityTypeManager()
    ->getStorage('taxonomy_term')
    ->loadTree($vocabulary_id);

  /** @var \Drupal\taxonomy\Entity\Term $term */
  foreach ($terms as $term) {
    $terms_array[] = [
      'id' => $term->tid,
      'name' => $term->name,
      'weight' => $term->weight,
    ];
  }
  $length = count($terms_array);
  $str = "Found $length terms in vocabulary $vocabulary_id<br>";

  foreach ($terms as $term) {
    $str .= 'Term: ' . $term->name . ' term_id: ' . $term->tid . '<br>';
  }
  $build['content'] = [
    '#type' => 'item',
    '#markup' => $str,
  ];
  return $build;
  }
}
```

Load all terms for a vocabulary and put them in a select (dropdown)

```
    $vid = 'event_format';
    $terms =\Drupal::entityTypeManager()->getStorage('taxonomy_term')->loadTree($vid);
    foreach ($terms as $term) {
      $options[$term->tid] = $term->name;
    }

    $form['event_format']['active'] = [
      '#type' => 'radios',
      '#title' => $this->t('Event Format'),
      '#default_value' => 1,
      '#options' => $options,
//      '#options' => [
//        0 => $this->t('In-Person'),
//        1 => $this->t('Online'),
//        2 => $this->t('In-Person & Online'),
//      ],
    ];
```

Create taxonomy term programatically

Vid is the vocabulary id e.g. event_category or type

```
$term = Term::create([
  'name' => 'protest',
  'vid' => 'event_category',
])->save();
```

Find all nodes with a matching term

See the queries chapter for other ways to do this.

```
public function getMatchingNodes() {
  $term_id = 3;

  $nodes = \Drupal::entityTypeManager()->getStorage('node')->loadByProperties([
    'field_event_category' => $term_id,
  ]);

  $length = count($nodes);
  $str = "Found $length nodes matching term_id $term_id<br>";

  /** @var Node $node */
  foreach ($nodes as $node) {
    $str .= "Node ". $node->id() . ": Title: " . $node->getTitle() . "<br>";
  }
  $build['content'] = [
    '#type' => 'item',
    '#markup' => $str,
  ];

  return $build;
}
```

Find nodes with a matching term using entityQuery

This finds the first 5 nodes that have the matching term.

```
protected function loadFirstOpinion($term_id) {
  $storage = \Drupal::entityTypeManager()->getStorage('node');
  $query = \Drupal::entityQuery('node')
    ->condition('status', 1)
    ->condition('type', 'opinion')
    ->condition('field_category', $term_id, '=')
    ->range(0, 5);
  $nids = $query->execute();
  $nodes = $storage->loadMultiple($nids);

  $ra = [];
  foreach ($nodes as $node) {
    $ra[] = [
      '#type' => 'markup',
      '#markup' => '<p>' . $node->getTitle(),
    ];
  }
  return $ra;
```

Back to top

Drupal at your fingertips by Selwyn Polit is licensed under CC BY 4.0

Page last modified: Apr 14 2023.

Edit this page on GitHub

This site uses Just the Docs, a documentation theme for Jekyll.