Test Case: no-orphans   FAIL   PASS

Test Scenario

Tagging three paragraphs, changing orders, and finally deleting one paragraph works as expecting: keeping tag/paragraph associations correct and removing the field tag entities as appropriate.

Pre-Conditions

1. An entity type with a paragraphs field.
2. The paragraphs field has field tagging enabled.
3. The paragraphs field allows unlimited items.

Test Data

tag1	alpha
tag2	bravo
tag3	charlie

Test Execution

Test Steps	Test Results
1. Begin adding new entity. 2. Create a copy paragraph with alpha as the value and tag it with alpha. 3. Create another copy paragraph with bravo as the value and tag it with bravo. 4. Save entity. 5. Reload the entity edit page.	FAIL PASS Assert the tags match their content.
6. Drag bravo before alpha 7. Save entity. 8. Reload the entity edit page.	FAIL PASS Assert the tags match their content.
9. Create a third copy paragraph with charlie as the value and tag it with charlie. 10. Drag it to the top position. 11. Save entity. 12. Reload the entity edit page.	FAIL PASS Assert all three tags match their content. FAIL PASS Assert the order is charlie, bravo, alpha.
13. Drag them in the original order: alpha, bravo, charlie. 14. Remove the paragraph with charlie. 15. Save entity. 16. Reload the entity edit page.	FAIL PASS Assert the edit form loads. FAIL PASS Assert all two tags match their content.

Test Steps	Test Results
17. Add a new third paragraph.	FAIL PASS Assert the tag field is empty, does not contain charlie

Comments

Please mention browser, device type, where you logged in, and anything else that can help diagnose failure.

Test completed by: Aaron Klump

Test Case: tags-in-code   FAIL   PASS

Test Scenario

It is possible to tag a field when creating and updating an entity programmatically; tags get attached correctly on read; field tag entities are deleted when the parent entity is deleted.

Pre-Conditions

1. An entity type with an image field called field_images
2. The image field has field tagging enabled.
3. The image field allows unlimited images.
4. An entity type with a paragraphs field called field_main_copy
5. The paragraphs field has field tagging enabled.
6. The paragraphs field allows unlimited items.
7. The snippets should be run in the *sandbox.inc* file.

Test Data

tag1	banner, vimeo
tag2	thumb image
tag3	foo bar
ptag1	apple
ptag2	banana

Test Execution

Test Steps	Test Results
1. Begin adding new entity. 2. Upload your first image and tag it banner, vimeo 3. Upload your second image and tag it thumb image 4. Create a copy paragraph with apple as the value and tag it with apple . 5. Create another copy paragraph with banana as the value and tag it with banana . 6. Create a third copy paragraph with any value and DO NOT tag it. 7. Save the entity. 8. Run *Snippet A* replacing $nid with the node ID of the entity you created.	FAIL PASS Assert the output is 'passed'

Test Steps	Test Results
9. Run *Snippet B* replacing $nid with the node ID of the entity you created.	FAIL PASS Assert the output is 'passed'; snippet B will fail on subsequent executions.
10. Reload the edit form for the entity.	FAIL PASS Assert the image is tagged with foo bar. FAIL PASS Assert field_main_copy has the correct tags in place.
11. Save the form. 12. Run *Snippet C* replacing $nid with the node ID of the entity you created.	FAIL PASS Assert the output is 'passed'; snippet C will fail on subsequent executions.
13. Reload the edit form for the entity.	FAIL PASS Assert the first image has the tag banner, vimeo FAIL PASS Assert there is no second image.
14. Delete the entity. 15. Run *Snippet D* replacing $nid with the node ID of the entity you created.	FAIL PASS Assert the output is 'passed'

Code Snippets

Snippet A

```php
<?php
$node = \Drupal\node\Entity\Node::load($nid);
$service = \Drupal::service('field_tag')->attachTags($node);

// Assert we can access a CSV tag.
$vimeos = $service->getItemsTaggedBy('vimeo', 'field_images');
1 === count($vimeos) || die('vimeo failed');

// Assert we can access another CSV tag.
$banners = $service->getItemsTaggedBy('banner', 'field_images');
1 === count($banners) || die('banner failed');

// Assert we can access a single tag with a space.
$thumbs = $service->getItemsTaggedBy('thumb image', 'field_images');
1 === count($thumbs) || die('thumb image failed');

// Assert the fieldTag instance.
// Assert the key is 1, which matches the delta of the field item.
$result = $thumbs[1]->fieldTag instanceof \Drupal\field_tag\Entity\FieldTag;
$result || die('no fieldTag instance');

die('Snippet A passed.');
```

Snippet B

```php
<?php
$node = \Drupal\node\Entity\Node::load( $nid );
$service = \Drupal::service('field_tag')->attachTags($node);
$i = $service->getItemsTaggedBy('thumb image', 'field_images');
$item = array_first($i)->getValue();
$item['field_tag'] = 'foo bar';
$node->field_images->filter(function () {
  return FALSE;
})->appendItem($item);
$node->save();
$service = \Drupal::service('field_tag')->attachTags($node);

$thumbs = $service->getItemsTaggedBy('thumb image', 'field_images');
0 === count($thumbs) || die('thumb image failed');

$foobars = $service->getItemsTaggedBy('foo bar', 'field_images');
1 === count($foobars) || die('foo bar failed');

$query = "select count(*) from ";
$query .= "field_tag where parent_id = $nid";
$count = \Drupal::database()->query($query)->fetchField();
4 == $count || die('field tag table should have 4 items');

die('Snippet B passed.');
```

Snippet C

```php
<?php
$node = \Drupal\node\Entity\Node::load( $nid );

$query = "select count(*) from ";
$query .= "field_tag where parent_id = $nid ";
$count = \Drupal::database()->query($query)->fetchField();
3 == $count || die('field tag table should have 3 items');

$item = $node->field_images->get(0)->getValue();
$node->field_images->filter(function () {
  return FALSE;
});

$item['field_tag'] = 'banner, vimeo';
$node->field_images->appendItem($item);
$item['field_tag'] = 'thumb image';
$node->field_images->appendItem($item);
$node->save();

$query = "select count(*) from ";
$query .= "field_tag where parent_id = $nid ";
$count = \Drupal::database()->query($query)->fetchField();
4 == $count || die('field tag table should have 4 items');

$item['field_tag'] = 'banner, vimeo';
$node->field_images->filter(function () {
  return FALSE;
})->appendItem($item);
$node->field_tag_sync[] = 'field_images';
$node->save();

$query = "select count(*) from ";
$query .= "field_tag where parent_id = $nid ";
$count = \Drupal::database()->query($query)->fetchField();
3 == $count || die('field tag table should have 3 items');

die('Snippet C passed.');
```

Snippet D

```php
<?php
$query = "select count(*) from ";
$query .= "field_tag where parent_id = $nid ";
$count = \Drupal::database()->query($query)->fetchField();
0 == $count || die('field tag table should have 0 items');

die('Snippet D passed.');
```

Comments

Please mention browser, device type, where you logged in, and anything else that can help diagnose failure.

Test completed by: Aaron Klump

Test Case: unsaved-multi-image   FAIL   PASS

Test Scenario

Tagging more than one image in a filefield works.

Pre-Conditions

1. An entity type with an image field.
2. The image field has field tagging enabled.
3. The image field allows unlimited images.

Test Data

tag1	alpha

Test Execution

Test Steps	Test Results
1. Begin adding new entity. 2. Upload your first image and tag it alpha 3. Upload your second image	FAIL PASS Assert alpha is still present in the first image's text field.

Comments

Please mention browser, device type, where you logged in, and anything else that can help diagnose failure.

Test completed by: Aaron Klump