Category Custom Image Attribute

Author: Damian Gambacorta

Objetivo

El documento tiene como propósito explicar los diferentes pasos que se deben realizar para poder crear un atributo en Category que permite subir imágenes custom.

1- Crear InstallData.php en [Vendor]\[Module]\Setup

```
<?php

namespace [Vendor]\[Module]\Setup;

use Magento\Eav\Setup\EavSetup;
use Magento\Eav\Setup\EavSetupFactory;
use Magento\Framework\Setup\InstallDataInterface;
use Magento\Framework\Setup\ModuleContextInterface;
use Magento\Framework\Setup\ModuleDataSetupInterface;
use Magento\Eav\Model\Entity\Attribute\ScopedAttributeInterface;

/**
 * Class InstallData
 * @package [Vendor]\[Module]\Setup
 */
class InstallData implements InstallDataInterface
{
      /**
      * @var EavSetupFactory
      */
      private $eavSetupFactory;

      /**
      * InstallData constructor.
      * @param EavSetupFactory $eavSetupFactory
      */
      public function __construct(EavSetupFactory $eavSetupFactory)
      {
      $this->eavSetupFactory = $eavSetupFactory;
      }

      /**
      * @param ModuleDataSetupInterface $setup
      * @param ModuleContextInterface $context
      */
      public function install(ModuleDataSetupInterface $setup,
ModuleContextInterface $context)
      {

      $eavSetup->addAttribute(
            \Magento\Catalog\Model\Category::ENTITY, 'image_name', [
            'type'      => 'varchar',
            'label'     => 'Image',
            'input'     => 'image',
            'backend' =>
'Magento\Catalog\Model\Category\Attribute\Backend\Image',
            'visible'  => true,
            'sort_order' => 9,
            'required' => false,
            'global'    =>
\Magento\Eav\Model\Entity\Attribute\ScopedAttributeInterface::SCOPE_STOR
E,
            'group'     => 'General Information',
      ]
      );

      $setup->endSetup();
      }
}
```

2-Crear el archivo category_form.xml en [Vendor]\[Module]\view\adminhtml\ui_component

```
<?xml version="1.0"?>
<form xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="urn:magento:module:Magento_Ui:etc/ui_conf
iguration.xsd">
      <fieldset name="content">
      <field name="image_name">
            <argument name="data" xsi:type="array">
                  <item name="config" xsi:type="array">
                  <item name="dataType" xsi:type="string">string</item>
                  <item name="source" xsi:type="string">category</item>
                  <item name="label" xsi:type="string"
translate="true">Label</item>
                  <item name="visible" xsi:type="boolean">true</item>
                  <item name="formElement"
xsi:type="string">fileUploader</item>
                  <item name="elementTmpl"
```

```
xsi:type="string">ui/form/element/uploader/uploader</item>
                  <item name="previewTmpl"
xsi:type="string">Magento_Catalog/image-preview</item>
                  <item name="required" xsi:type="boolean">false</item>
                  <item name="sortOrder" xsi:type="number">40</item>
                  <item name="uploaderConfig" xsi:type="array">
                        <item name="url" xsi:type="url"
path="[module]/category_image/upload"/>
                  </item>
                  </item>
            </argument>
      </field>
      </fieldset>
</form>
```

3-Agregar el siguiente codigo en [Vendor]/[Module]/etc/di.xml

```
<virtualType name="Magento\Catalog\CategoryImageUpload"
type="Magento\Catalog\Model\ImageUploader">
      <arguments>
            <argument name="baseTmpPath"
xsi:type="string">catalog/tmp/category</argument>
            <argument name="basePath"
xsi:type="string">catalog/category</argument>
            <argument name="allowedExtensions" xsi:type="array">
                  <item name="jpg" xsi:type="string">jpg</item>
                  <item name="jpeg" xsi:type="string">jpeg</item>
                  <item name="gif" xsi:type="string">gif</item>
                  <item name="png" xsi:type="string">png</item>
            </argument>
      </arguments>
      </virtualType>
      <type
name="[Vendor]\[Module]\Controller\Adminhtml\Category\Image\Upload">
      <arguments>
            <argument name="imageUploader"
xsi:type="object">Magento\Catalog\CategoryImageUpload</argument>
      </arguments>
      </type>
      <preference for="Magento\Catalog\Model\Category\DataProvider"
type="[Vendor]\[Module]\Model\Category\DataProvider" />
```

4- Crear Upload.php file on location [Vendor]/[Module]/Controller/Adminhtml/Category/Image

```php
<?php
namespace [Vendor]\[Module]\Controller\Adminhtml\Category\Image;
use Magento\Framework\Controller\ResultFactory;

/**
 * Class Upload
 * @package [Vendor]\[Module]\Controller\Adminhtml\Category\Image
 */
class Upload extends \Magento\Backend\App\Action
{
      /**
      * @var \Magento\Catalog\Model\ImageUploader
      */
      protected $imageUploader;

      /**
      * Upload constructor.
      * @param \Magento\Backend\App\Action\Context $context
      * @param \Magento\Catalog\Model\ImageUploader $imageUploader
      */
      public function __construct(
      \Magento\Backend\App\Action\Context $context,
      \Magento\Catalog\Model\ImageUploader $imageUploader
      ) {

      parent::__construct($context);
      $this->imageUploader = $imageUploader;
      }

      /**
      * @return bool
      */
      protected function _isAllowed()
      {
      return
$this->_authorization->isAllowed('Magento_Catalog::categories');
      }

      /**
      * @return
\Magento\Framework\App\ResponseInterface|\Magento\Framework\Controller\R
esultInterface
      */
      public function execute()
      {
```

```
      try {

            $result =
$this->imageUploader->saveFileToTmpDir('image_name');

            $result['cookie'] = [
                  'name'  => $this->_getSession()->getName(),
                  'value'     => $this->_getSession()->getSessionId(),
                  'lifetime'    =>
$this->_getSession()->getCookieLifetime(),
                  'path'      => $this->_getSession()->getCookiePath(),
                  'domain'    => $this->_getSession()->getCookieDomain(),
            ];
      } catch (\Exception $e) {

            $result = ['error' => $e->getMessage(), 'errorcode' =>
$e->getCode()];
      }
      return
$this->resultFactory->create(ResultFactory::TYPE_JSON)->setData($result)
;
      }
}
```

5- Crear DataProvider.php file on location [Vendor]\[Module]\Model\Category

```
<?php
namespace [Vendor]\[Module]\Model\Category;

/**
 * Class DataProvider
 * @package [Vendor]\[Module]\Model\Category
 */
class DataProvider extends \Magento\Catalog\Model\Category\DataProvider
{
      /**
      * @return array
      */
      protected function getFieldsMap()
      {
      $fields = parent::getFieldsMap();
      $fields['content'][] = 'image_name';

      return $fields;
      }
}
```

6- Crear en etc/adminhtml el archivo routes.xml

```
<?xml version="1.0" ?>
<config xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="urn:magento:framework:App/etc/routes.xsd"
>
<router id="admin">
      <route frontName="[module]" id="[module]">
      <module name="[Vendor]_[Module]" before="Magento_Backend"/>
      </route>
</router>
</config>
```

7- Es posible, que cuando se indique un mensaje de error en el formulario de carga, la imagen nueva agregada se rompa, debido a un preprocesamiento de la imagen que hace magento en el backend. El siguiente workaround soluciona el problema.
Crear en Controller/Catalog/Adminhtml/Category/Save.php

```
<?php

namespace [Vendor]\[Module]\Controller\Catalog\Adminhtml\Category;

/**
* Class Save
* @package [Vendor]\[Module]\Controller\Catalog\Adminhtml\Category
*/
class Save extends
\Magento\Catalog\Controller\Adminhtml\Category\Save
{

   /**
    * Workaround
    * Solve a bug, that when user try to save a category with footer
option,
    * and max categories allowed already satisfied, the icon image
show wrong
    * preview.
    *
    * @param array $data
    * @return array
```

```
     */

    public function imagePreprocessing($data)
    {

        $dataLocal = parent::imagePreprocessing($data);

        if ($dataLocal['image_name'] == false) {
            $dataLocal['image_name'] = null;
        }

        return $dataLocal;
    }
```

}