# Setup Scripts

…/Setup/InstallSchema.php

Thiết lập DB,

**<?php namespace** Manager\Banner\Setup;  
**use** Magento\Framework\Setup\InstallSchemaInterface;  
**use** Magento\Framework\Setup\ModuleContextInterface;  
**use** Magento\Framework\Setup\SchemaSetupInterface;  
**use** Magento\Framework\DB\Ddl\Table;  
**class** InstallSchema **implements** InstallSchemaInterface  
{  
 **public function** install(SchemaSetupInterface $setup, ModuleContextInterface $context)  
 {  
 $setup->startSetup();  
 $connection = $setup->getConnection();  
 $tableName = $setup->getTable("banner");  
 **if**($connection->isTableExists($tableName) != **true**)  
 {  
 //identity: automatic increment  
 $table = $connection->newTable($tableName)->addColumn(  
 "id",  
 Table::*TYPE\_INTEGER*,  
 **null**,  
 ["primary" => **true**, "nullable"=>**false**, "identity"=>**true**]  
 )->addColumn(  
 "title",  
 Table::*TYPE\_TEXT*,  
 255,  
 ["nullable"=>**false**]  
 )->addColumn(  
 "content",  
 Table::*TYPE\_TEXT*,  
 255,  
 ["nullable"=>**false**]  
 )->addColumn(  
 "image",  
 Table::*TYPE\_TEXT*,  
 255,  
 ["nullable"=>**false**]  
 )->addColumn(  
 "link",  
 Table::*TYPE\_TEXT*,  
 255,  
 ["nullable"=>**false**]  
 )->addColumn(  
 "status",  
 Table::*TYPE\_SMALLINT*,  
 2,  
 ["nullable"=>**false**]  
 )->addColumn(  
 'creation\_time',  
 Table::*TYPE\_TIMESTAMP*,  
 **null**,  
 ['nullable' => **false**, 'default' => Table::*TIMESTAMP\_INIT*],  
 'Block Creation Time')  
 ->addColumn(  
 'update\_time',  
 Table::*TYPE\_TIMESTAMP*,  
 **null**,  
 ['nullable' => **false**, 'default' => Table::*TIMESTAMP\_INIT\_UPDATE*],  
 'Block Modification Time'  
 )->setOption("charset","utf8");  
  
 $connection->createTable($table);  
 }  
 $setup->endSetup();  
 }  
}

# Model

* Model : chứa các phương thức được định nghĩa

**<?php  
namespace** Manager\Banner\Model;  
**class** Banner **extends** \Magento\Framework\Model\AbstractModel  
{  
  
 **const** *STATUS\_ENABLED* = 1;  
 **const** *STATUS\_DISABLED* = 0;  
 **protected function** \_construct()  
 {  
 $this->\_init('Manager\Banner\Model\ResourceModel\Banner');  
 }  
  
 **public function** getAvailableStatuses()  
 {  
 **return** [**self**::*STATUS\_ENABLED* => \_\_('Enabled'), **self**::*STATUS\_DISABLED* => \_\_('Disabled')];  
 }  
}

* ResourceModel: tương tác với database

**<?php  
namespace** Manager\Banner\Model\ResourceModel;  
**class** Banner **extends** \Magento\Framework\Model\ResourceModel\Db\AbstractDb  
{  
 **protected function** \_construct()  
 {  
 // Table name + primary key column  
 $this->\_init('banner', 'id');  
 }  
 **protected function** \_afterSave(\Magento\Framework\Model\AbstractModel $object)  
 {  
 // Get image data before and after save  
 $oldImage = $object->getOrigData('image');  
 $newImage = $object->getData('image');  
 // Check when new image uploaded  
 **if** ($newImage != **null** && $newImage != $oldImage) {  
 $imageUploader = \Magento\Framework\App\ObjectManager::*getInstance*()  
 ->get('Manager\Banner\BannerImageUpload');  
 $imageUploader->moveFileFromTmp($newImage);  
 }  
 **return** $this;  
 }  
  
}

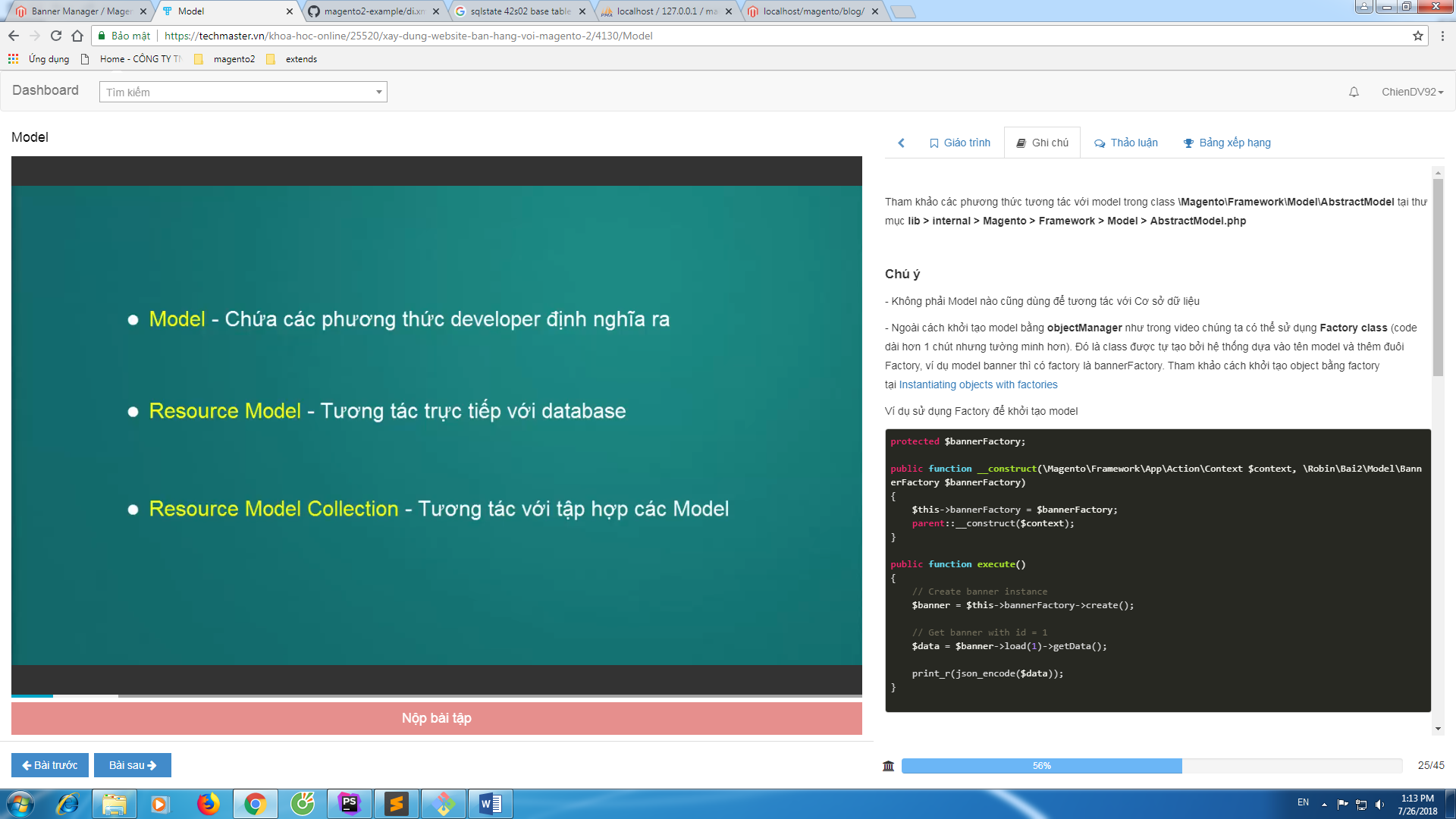
* Collection: kết nối giữa Model và ResourceModel (tập hợp các model)

**<?php  
namespace** Manager\Banner\Model\ResourceModel\Banner;  
**class** Collection **extends** \Magento\Framework\Model\ResourceModel\Db\Collection\AbstractCollection  
{  
 **protected** $\_idFieldName = 'id';  
 **protected function** \_construct()  
 {  
 // Model + Resource Model  
 $this->\_init('Manager\Banner\Model\Banner', 'Manager\Banner\Model\ResourceModel\Banner');  
 }  
}

Khởi tạo Model

* **objectManager**
* **Factory class** ( Đó là class được tự tạo bởi hệ thống dựa vào tên model và thêm đuôi Factory, ví dụ model banner thì có factory là bannerFactory)\

**Ví dụ:**



# Sử dụng collection

Controller (sử dụng collection để lấy data, có them điều kiện)

* Khởi tạo collection

**$collection = $this->bannerFactory->create()->getCollection();**

* Lấy data theo collection

|  |
| --- |
| Ex: |
|  |  |

**SELECT \* FROM banner**

$data = $collection->getData();

**SELECT \* FROM banner WHERE id > 50 (gt = greater than (lớn hơn)) addFieldToFilter(bổ xung thêm điều kiện where)**

$data = $collection->addFieldToFilter('id', ['gt' => 50])->getData();

**SELECT id FROM banner WHERE id > 50 (addFieldToSelect(lựa chọn tên cột cần lấy))**

$data = $collection->addFieldToSelect('id')

->addFieldToFilter('id', ['gt' => 50])->getData();

**SELECT \* FROM banner WHERE image LIKE '%.png'**

$data = $collection->addFieldToFilter('image', ['like' => '%.png'])->getData();

print\_r(json\_encode($data));

**SELECT \* FROM banner WHERE image LIKE '%.png' AND id > 50**

$query = $collection->addFieldToFilter('image', ['like' => '%.png'])

->addFieldToFilter('id', ['gt' => 50])

->getSelect();

**SELECT \* FROM banner WHERE (image LIKE '%.png' OR image LIKE '%.jpg') AND id > 50**

$query = $collection->addFieldToFilter('id', ['gt' => 50])

->addFieldToFilter('image',

[

['like' => '%.png'],

['like' => '%.jpg']

])->getSelect();

**SELECT \* FROM banner WHERE id > 50 OR image LIKE '%.jpg'**

$query = $collection->addFieldToFilter(['id', 'image'],

[

['gt' => 50],

['like' => '%.png']

])->getSelect();

echo $query;

**Cấu trúc mảng điều kiện của Magento**

| **Cú pháp** | **Ý nghĩa** |
| --- | --- |
| ["**from**" => $fromValue, "to" => $toValue] | $column **FROM** $fromValue **TO**$toValue |
| ["**eq**" => $value] | $column **=**$value |
| ["**neq**" => $value] | $column **!=** $value |
| ["**like**" => $value] | $column **LIKE** $value |
| ["**in**" => [$value]] | $column **IN** ($values) |
| ["**nin**" => [$value]] | $column **NOT IN** ($values) |
| ["**notnull**" => $value] | $column **IS NOT NULL** |
| ["**null**" => $value] | $column **IS NULL** |
| ["**moreq**" => $value] | **(**$column **=** $value**)** |
| ["**gt**" => $value] | $column **>** $value |
| ["**lt**" => $value] | $column **<** $value |
| ["**gteq**" => $value] | $column **>=** $value |
| ["**lteq**" => $value] | $column **<=** $value |
| ["**finset**" => $value] | **FIND\_IN\_SET**($value, $column) |
| ["**regexp**" => $value)] | $column **REGEXP** $value |
| ["**seq**" => $value] | $column **=** $value |
| ["**sneq**" => $value] | $column **!=** $value |

# Module

## 1 - Backend Router

**File /etc/adminhtml/routes.xml (điều hướng cho module)**

<?xml version="1.0"?>  
<!--  
/\*\*  
 \* Copyright © Magento, Inc. All rights reserved.  
 \* See COPYING.txt for license details.  
 \*/  
-->  
<config xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="urn:magento:framework:App/etc/routes.xsd">  
 <router id="admin">  
 <route id="banner" frontName="banner">  
 <module name="Manager\_Banner" before="Magento\_Backend" />  
 </route>  
 </router>  
</config>

**Thẻ router**

**Id = admin : cho backend**

**Id = standard : cho fontend**

**Thẻ route thường để id = frontName**

* **Controller** load layout and set active menu

|  |  |
| --- | --- |
|  | $resultPage = $this->resultPageFactory->create(); |
|  | $resultPage->setActiveMenu('Manager\_Banner::banner\_manager'); // active menu |
|  | $resultPage->getConfig()->getTitle()->prepend(\_\_('Banner manager'));// lấy title |
|  |  |
|  | return $resultPage; |

## 2 - Backend Menu

**/etc/adminhtml/menu.xml tạo link cho module trong menu admin**

<?xml version="1.0"?>  
  
<config xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="urn:magento:module:Magento\_Backend:etc/menu.xsd">  
 <menu>  
 <add id="Manager\_Banner::manager" title="Manager" translate="title" module="Manager\_Banner" sortOrder="10" resource="Manager\_Banner::manager"/>  
 <add id="Manager\_Banner::banner\_manager" title="Banner" translate="title" module="Manager\_Banner" sortOrder="60"  
 parent="Manager\_Banner::manager" action="banner/index/index" resource="Manager\_Banner::banner"/>  
 </menu>  
</config>

Menu : để khai báo menu trong admin

Add : them file (label, link)

* Id:  Mã của menu item, chỉ chấp nhận các ký tự theo pattern sau: [A-Za-z0-9/\_:]{3,}
* Title: tiêu đề menu
* ToolTip : Text hiển thị khi hover lên menu 1 lúc (thuộc tính **title** trong HTML)
* Module: tên module
* Sortorder: Vị trí sắp xếp so với các menu item khác. Giá trị là 1 số nguyên, số càng nhỏ thì càng được ưu tiên lên đầu
* Parent: kế thừa module nào (link nằm trong menu của module nào) Menu cha. Giá trị là id của menu item khác
* Action: Đường dẫn khi click vào menu item
* Resource : Mã resource dùng trong phân quyền (ACL)

**Chú ý**

Để xóa hoặc cập nhật một menu item, ta sử dụng element <remove> hoặc <update> thay vì element <add>. Ví dụ xóa menu **Content**:

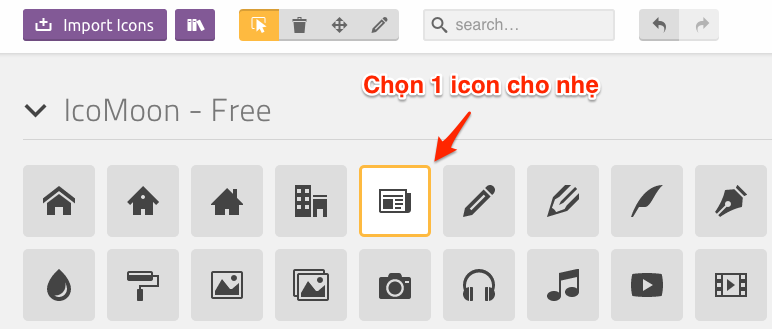
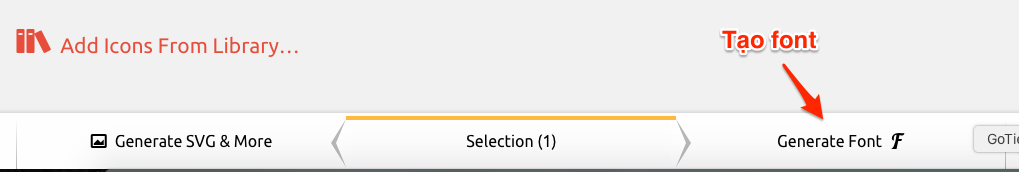
*<remove id*="Magento\_Backend::content"/>

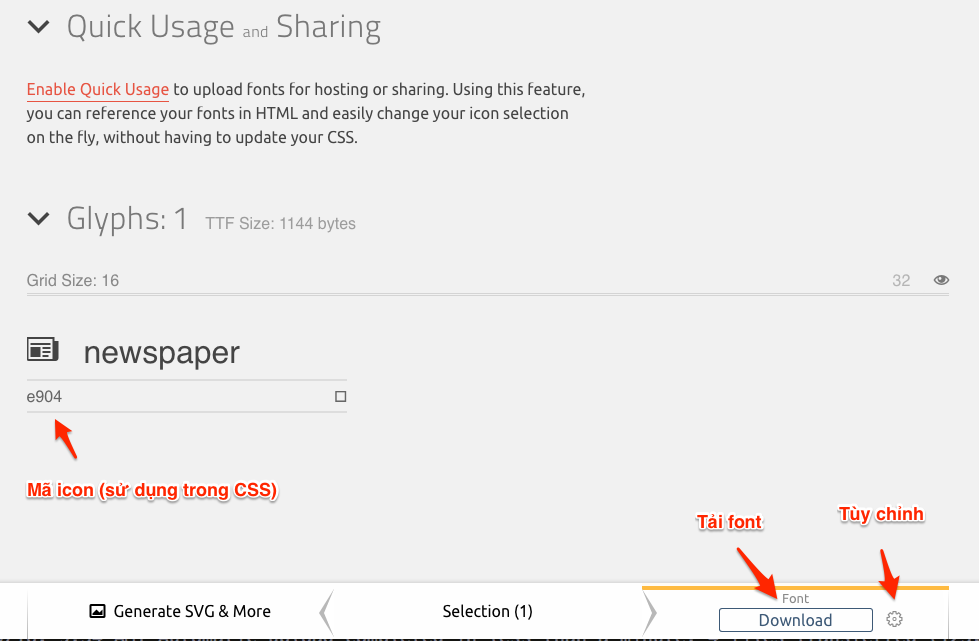
**Cách tạo custom icon cho menu**

**1. Tạo icon**

- Tạo 1 icon vector bằng phần mềm đồ họa (đuôi file .svg) sau đó convert sang font sử dụng trang <https://icomoon.io/> (hoặc có thể sử dụng luôn các font có sẵn). Ví dụ sử dụng font có sẵn:

Bật IcoMoon App

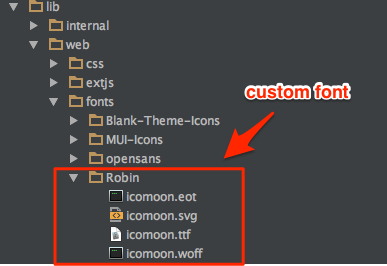
Chọn iconTạo font từ icon đã chọn

Tải font về máy

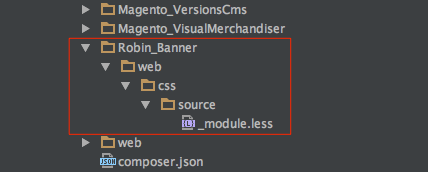
**2. Nhúng font vào source code**

- Tạo 1 thư mục ở trong **lib > web > fonts**(ví dụ Manager)

- Copy file font vào trong thư mục trên (nếu là font convert từ IcoMoon thì gồm 4 file nằm trong thư mục fonts)



- Trong thư mục **app > design > adminhtml > Magento > backend** tạo một thư mục để cập nhật css cho module (ví dụ Manager\_Banner). Sau đó tạo file \_module.less theo cấu trúc sau:



Nội dung file **\_module.less**(không tính comment)

@*Manager*-*icons*-*admin\_\_font*-*name*-*path*: '@{baseDir}fonts/Manager/icomoon'; /\* Đường dẫn đến file fonts (không có đuôi) \*/  
@*Manager*-*icons*-*admin\_\_font*-*name* : 'Manager'; /\* Font name tự đặt \*/  
.*lib*-*font*-face(  
 @*family*-*name*:@*Manager*-*icons*-*admin\_\_font*-*name*,  
 @*font*-*path*: @*Manager*-*icons*-*admin\_\_font*-*name*-*path*,  
 @*font*-*weight*: *normal*,  
 @*font*-*style*: *normal*);  
.*admin\_\_menu* .*item*-*banner*-*manager*.*parent*.*level*-0 > *a*:*before* { /\* Chú ý item-banner-manager là class sinh ra dựa vào id module \*/  
 *font*-*family*: @*Manager*-*icons*-*admin\_\_font*-*name*;  
 content: "\e90d"; /\* Mã của icon \*/  
}

**3. Cập nhật lại CSS bằng cách chạy lệnh Magento CLI**

Chạy lệnh sau ở trong thư mục gốc của project:

$ sudo rm -rf **var**/view\_processing/

$ sudo rm -rf **var**/generation/

$ sudo rm -rf pub/**static**/adminhtml/

$ sudo bin/magento setup:**static**-content:deploy

**4. Bổ sung file font còn thiếu để support được nhiều trình duyệt, thiết bị hơn**

- Font trong Magento thường sẽ có 5 định dạng, nếu là font được convert bằng <https://icomoon.io/> thì sẽ thiếu 1 định dạng là **.woff2**

- Có thể vào trang [fontsquirrel.com](http://www.fontsquirrel.com/tools/webfont-generator) để convert font sang các định dạng còn thiếu, sau đó lặp lại bước 2 và 3 để cập nhật lại font.

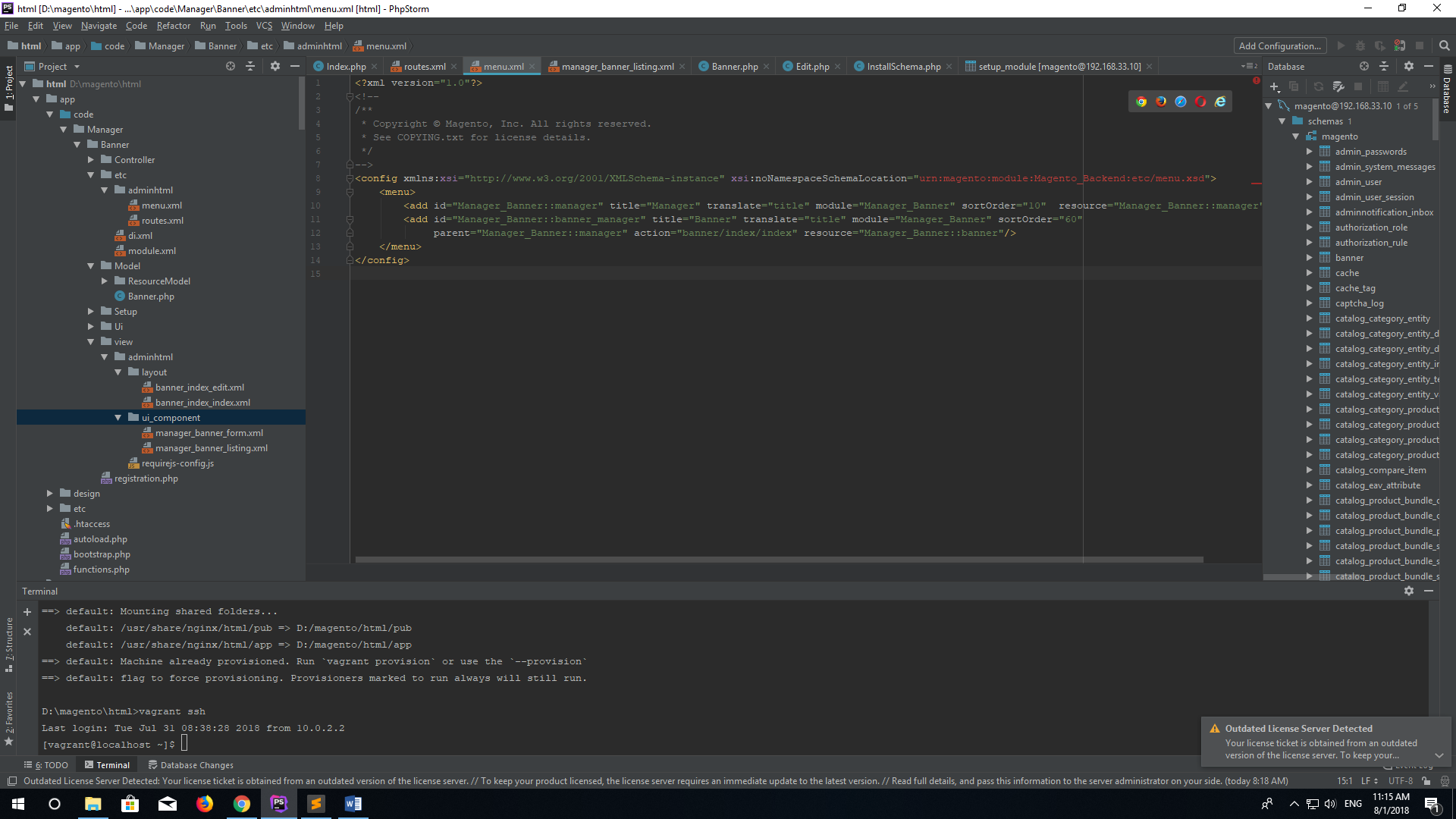
## 3 - Listing

Hiển thị danh sách

* Sử dụng ui\_component (render UI bằng file xml) https://devdocs.magento.com/guides/v2.3/ui\_comp\_guide/bk-ui\_comps.html
* /etc/di.xml

**Dependency injection cho các module hoặc tạo các class con**

* Tạo view/adminhtml



* **Layout**

/view/admin/layout/banner\_index\_edit.xml (dung cho controller index (controller/adminhtml/index /index) )

<?xml version="1.0"?>  
  
<page xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="urn:magento:framework:View/Layout/etc/page\_configuration.xsd">  
 <update handle="styles"/>  
 <body>  
 <referenceContainer name="content">  
 <uiComponent name="manager\_banner\_listing"/>  
 </referenceContainer>  
 </body>  
</page>

dung ui với name = manager\_banner\_listing tức là gọi đến file manager\_banner\_listing.xml nằm trong /view/adminhtml/ui\_component

* Sửa lại các class, thuộc tính, colums cho tương ứng để hiển thị data ra ngoài giao diện

/etc/di.xml

* **Để lại element type(CollectionFactory)**

Index tham số manager\_banner\_listing\_data\_source vào trong class collectionFactory trong Banner\Grid\Collection và được khai báo ở dưới

<type name="Magento\Framework\View\Element\UiComponent\DataProvider\CollectionFactory">  
 <arguments>  
 <argument name="collections" xsi:type="array">  
 <item name="manager\_banner\_listing\_data\_source" xsi:type="string">Manager\Banner\Model\ResourceModel\Banner\Grid\Collection</item>  
 </argument>  
 </arguments>  
</type>

* **Để lại type (/Grid/Collection)**

Sử dung virtualType với thuộc tính type =… tức là tạo 1 sub class của class dataprovider

Bên trong chứa các contructor class

* + mainTable
  + resourceModel

<virtualType name="Manager\Banner\Model\ResourceModel\Banner\Grid\Collection" type="Magento\Framework\View\Element\UiComponent\DataProvider\SearchResult">  
 <arguments>  
 <argument name="mainTable" xsi:type="string">banner</argument>  
 <argument name="resourceModel" xsi:type="string">Manager\Banner\Model\ResourceModel\Banner</argument>  
 </arguments>  
</virtualType>

**Một số element chính trong Grid Component**

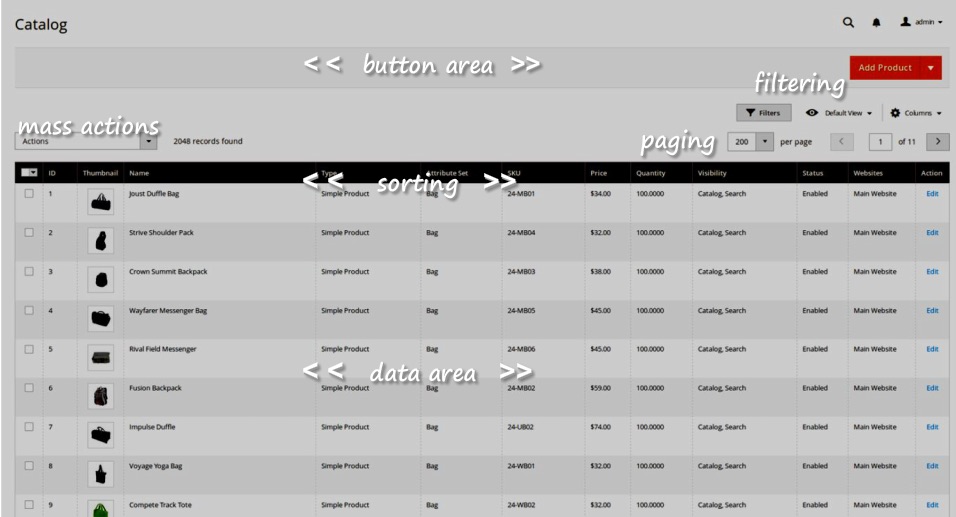
- **dataSource**: Nguồn dữ liệu đổ ra bảng

- **listingToolbar**: Thanh toolbar trên cùng, chứa các thành phần như:

* **bookmark**: Lưu các view khác nhau khi search, filter hoặc ẩn columns
* **columnsControls**: Cho phép chọn/ẩn cột hiển thị
* **filterSearch**: Khung search
* **filters**: Lọc dữ liệu
* **massaction**: Hành động tác dụng lên các dòng dữ liệu đang chọn
* **paging**: Phân trang

- **columns**: Cấu hình các cột trong bảng:

* **selectionsColumn**: Cột selections (checkbox), dùng kết hợp với Mass action
* **column**: Cấu hình các cột
* **actionsColumn**: Cột actions, chứa các link đến các action như edit, delete, ...



**Custom select column**

Cột **Status** có kiểu dữ liệu là **boolean**, khi hiển thị trên danh sách toàn **0** và **1** rất khó hiểu => Sửa lại nếu là **0** thì hiển thị **Disable**, nếu là **1** thì hiển thị **Enable**, trong phần Filter cũng sửa lại thành select cho lựa chọn giữa **Enable**và **Disable**(tham khảo module *Cms*)

- Sửa cấu hình cột **status** trong **banner\_listing.xml**(sửa *datatype* và *options*):

<*column name*="status">  
 <*argument name*="data" xsi:*type*="array">  
 <*item name*="options" xsi:*type*="object">Manager\Banner\Model\Banner\Source\*Status*</*item*>  
 <*item name*="config" xsi:*type*="array">  
 <*item name*="filter" xsi:*type*="string">*select*</*item*>  
 <*item name*="component" xsi:*type*="string">*Magento\_Ui*/*js*/*grid*/*columns*/*select*</*item*>  
 <*item name*="editor" xsi:*type*="string">*select*</*item*>  
 <*item name*="dataType" xsi:*type*="string">*select*</*item*>  
 <*item name*="label" xsi:*type*="string" *translate*="true">*Status*</*item*>  
 </*item*>  
 </*argument*>  
</*column*>

- Tạo Class dùng để render ra Options giống khai báo ở trên

// File [Module]/Model/Banner/Source/Status.php  
  
<?php  
**namespace** Manager\Banner\Model\Banner\Source;  
**class** Status **implements** \Magento\Framework\Data\OptionSourceInterface  
{  
 **protected** $banner;  
 **public function** \_\_construct(\Manager\Banner\Model\Banner $banner)  
 {  
 $this->banner = $banner;  
 }  
 **public function** toOptionArray()  
 {  
 $availableOptions = $this->banner->getAvailableStatuses();  
 $options = [];  
 **foreach** ($availableOptions **as** $key => $value) {  
 $options[] = [  
 'label' => $value,  
 'value' => $key,  
 ];  
 }  
 **return** $options;  
 }  
}

- Bổ sung vào Model **Banner** phương thức lấy ra mảng các giá trị của Status và label của chúng:

**const** *STATUS\_ENABLED* = 1;  
**const** *STATUS\_DISABLED* = 0;  
  
**public function** getAvailableStatuses()  
{  
 **return** [**self**::*STATUS\_ENABLED* => \_\_('Enabled'), **self**::*STATUS\_DISABLED* => \_\_('Disabled')];  
}

## 4 - Forward & Redirect

Tạo New và edit (tạo chung 1 form vì giống nhau).

* Tạo file Newaction.php and Edit.php trong /Controller/Adminhtml/Index/
  + Newaction.php

$resultForward = $this->resultForwardFactory->create();  
**return** $resultForward->forward('edit');

* + - Chuyển hướng trang
  + Edit.php
    - Có biến registry dung để đăng kí 1 biến có thể dùng được bên view

**protected function** \_initAction()  
{  
 // load layout, set active menu and breadcrumbs  
 */\*\** ***@var*** *\Magento\Backend\Model\View\Result\Banner $resultBanner \*/* $resultBanner = $this->resultBannerFactory->create();  
 $resultBanner->setActiveMenu('Manager\_Banner::manager')  
 ->addBreadcrumb(\_\_('CMS'), \_\_('CMS'))  
 ->addBreadcrumb(\_\_('Manage Banners'), \_\_('Manage Banners'));  
 **return** $resultBanner;  
}

set active menu

**public function** execute()  
{  
 // Get ID and create model  
 $id = $this->getRequest()->getParam('id');  
 $model = $this->\_objectManager->create('Manager\Banner\Model\Banner');  
 // Initial checking  
 **if** ($id) {  
 $model->load($id);  
 // If cannot get ID of model, display error message and redirect to List page  
 **if** (!$model->getId()) {  
 $this->messageManager->addError(\_\_('This image no longer exists.'));  
 $resultRedirect = $this->resultRedirectFactory->create();  
 **return** $resultRedirect->setPath('\*/\*/');  
 }  
 }  
 // Registry banner  
 $this->\_coreRegistry->register('banner', $model);  
 // Build form  
 $resultPage = $this->\_initAction();  
 $resultPage->getConfig()->getTitle()  
 ->prepend($model->getId() ? $model->getImage() : \_\_('Create Image'));  
 **return** $resultPage;  
}

* Note
  + Form ui chú ý

dataSource

dataProvider

phải gọi đúng class

Action Delete

$this->\_eventManager->dispatch(  
 'adminhtml\_cmspage\_on\_delete',  
 ['title' => $title, 'status' => 'success']  
);

dùng để tạo ra 1 **event** (sự kiện), các module khác có thể bắt được event đó xảy ra để thực hiện 1 tác vụ nào đó (ví dụ trong trường hợp này là sự kiện xóa 1 trang). Tham khảo thêm về cách sử dụng event trong Magento 2 tại đây [Events and observers](http://devdocs.magento.com/guides/v2.1/extension-dev-guide/events-and-observers.html)

## 5 – ACL (phân quyền) Access Control List

Tạo file /etc/acl.xml (file để tạo quyền trong admin)

Sau khi cấu hình ACL xong cần sửa lại quyền truy cập vào các action tương ứng với các quyền đã định nghĩa:

- Ghi đè hằng số **ADMIN\_RESOURCE** có giá trị bằng quyền đã định nghĩa. Ví dụ ở action Edit gán với quyền **Manager\_Banner::save**

**const** *ADMIN\_RESOURCE* = 'Manager\_Banner::save';

- Sửa lại **resource** của menu cho phù hợp. Ví dụ menu sau chỉ xuất hiện khi user có quyền **Manager\_Banner::save**

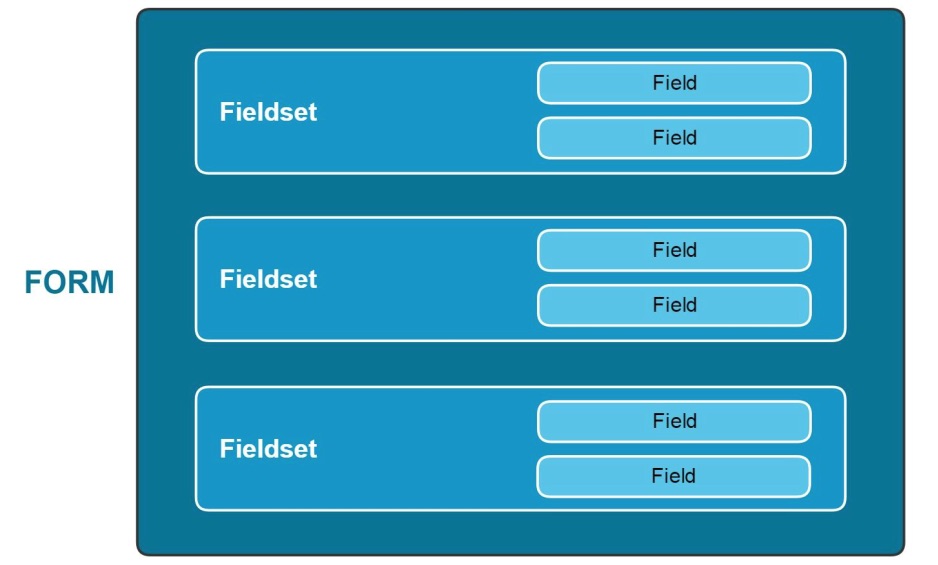
<*add id*="Manager\_Banner::new\_banner" *title*="New Banner" *module*="Manager\_Banner" *action*="banner/index/new" *parent*="Manager\_Banner::banner\_manager" *resource*="Manager\_Banner::save"/>

## 6 – Form

* Tạo file /view/adminhtml/layout/banner\_index\_edit.xml
* <?xml version="1.0"?>  
  <!--  
  /\*\*  
   \* Copyright © Magento, Inc. All rights reserved.  
   \* See COPYING.txt for license details.  
   \*/  
  -->  
  <page xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" layout="admin-1column" xsi:noNamespaceSchemaLocation="urn:magento:framework:View/Layout/etc/page\_configuration.xsd">  
   <update handle="editor"/>  
   <body>  
   <referenceContainer name="content">  
   <uiComponent name="manager\_banner\_form"/>  
   </referenceContainer>  
   </body>  
  </page>
* Trong thẻ uicomponent có thuộc tính name="manager\_banner\_form" gọị đến file /view/adminhtml/ui\_component/manager\_banner\_form.xml
* Trong ui-component/manager\_banner\_form.xml cần chỉnh sửa các class và các field cho đúng

Tài liệu Tham khảo thêm về Form component:

<http://devdocs.magento.com/guides/v2.1/ui-components/ui-form.html>



**Chú ý**

- Thẻ <update> trong layout XML dùng để chèn 1 file layout khác với tên là giá trị của thuộc tính handle. Ví dụ:

<*update handle*="editor"/>

tức là chèn file **editor.xml**vào trong layout (có thể dùng chức năng **Navigate / File ...** trong PHPStorm để tìm kiếm file).

Thông thường trang **Listing** nên dùng **handle="styles"**, trang **New** và **Edit** nên dùng **handle="editor".**

- Để validate Form dùng thư viện Javascript có sẵn thì khai báo trong thẻ <item name="validation">, ví dụ validate **required-entry:**

<*item name*="validation" xsi:*type*="array">  
 <*item name*="required-entry" xsi:*type*="boolean">**true**</*item*>  
</*item*>

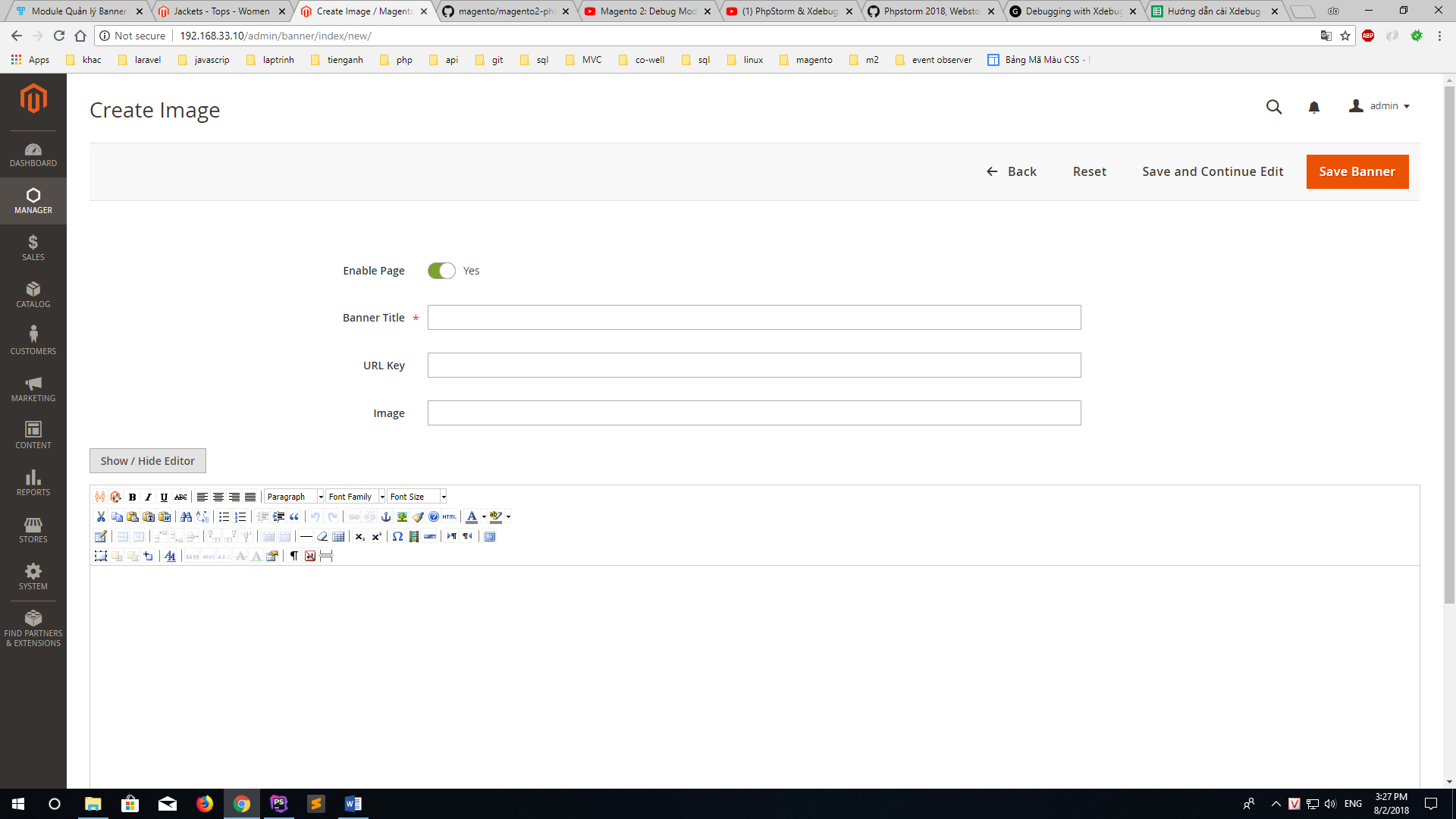
Một số kiểu validate phải nhập thêm giá trị, ví dụ validate **min\_text\_length** phải nhập số ký tự tối thiểu:

<*item name*="validation" xsi:*type*="array">  
 <*item name*="min\_text\_length" xsi:*type*="number">5</*item*>  
</*item*>

- Tham khảo validate form bằng Javascript trong file**/app/code/Magento/UI/view/base/web/js/lib/validation/rule.js**

**- Tạo các nút và gọi class cho các nut**

**Gọi đúng class ở ui form**



* **Tạo datasource**

Sửa name datasource

Sửa id

Sửa path active save để lưu database submitUrl path="banner/save"/>

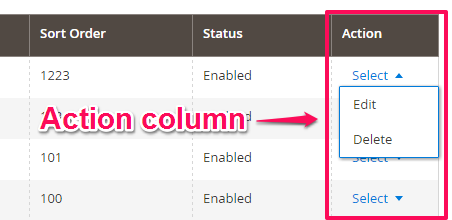
Tạo Class dataprovider (nó là 1 model)

<dataSource name="banner\_form\_data\_source">  
 <argument name="data" xsi:type="array">  
 <item name="js\_config" xsi:type="array">  
 <item name="component" xsi:type="string">Magento\_Ui/js/form/provider</item>  
 </item>  
 </argument>  
 <settings>  
 <submitUrl path="banner/save"/>  
 </settings>  
 <dataProvider class="Manager\Banner\Model\Banner\DataProvider" name="banner\_form\_data\_source">  
 <settings>  
 <requestFieldName>id</requestFieldName>  
 <primaryFieldName>id</primaryFieldName>  
 </settings>  
 </dataProvider>  
</dataSource>

## 7 - Data Processor

* Xử lý action save cho new và edit
* /Controller/Adminhtml/Index/Save.php
* Tạo file /Controller/Adminhtml/Index/PotsDataProcessor.php (để validate chứ kp là action)
* **Tổng hợp 1 số cấu hình validate bằng Javascript trong form**

|  |  |
| --- | --- |
| **Tên** | **Tác dụng** |
| required-entry | Không được để trống |
| no-whitespace | Không được để trống, không được nhập khoảng trắng |
| validate-integer | Nhận giá trị số nguyên, được phép để trống |
| validate-number | Nhận giá trị số (cả số âm và số thập phân), được phép để trống |
| validate-digits | Nhận các ký tự số **0-9**, được phép để trống |
| validate-alpha | Nhận các ký tự chữ cái **a-z** hoặc **A-Z**, được phép để trống |
| validate-alphanum | Nhận các ký tự chữ số **a-z**, **A-Z** và **0-9**, được phép để trống |
| validate-code | Nhận các ký tự chữ số **a-z**, **A-Z**, **0-9** và dấu gạch dưới **\_**, được phép để trống |
| validate-emails | Nhận giá trị địa chỉ email hợp lệ, được phép để trống |
| validate-date | Nhận giá trị ngày tháng năm (giờ, phút, giây) hợp lệ, được phép để trống |

* **Hướng dẫn cấu hình Action column trang List**
* 

- Cấu hình class xử lý logic để tạo dữ liệu cho Action column (file banner\_listing.xml):

<actionsColumn name="actions" **class**="Manager\Banner\Ui\Component\Listing\Column\Actions">  
 <argument name="data" xsi:type="array">  
 <item name="config" xsi:type="array">  
 <item name="resizeEnabled" xsi:type="boolean">false</item>  
 <item name="resizeDefaultWidth" xsi:type="string">107</item>  
 <item name="indexField" xsi:type="string">id</item>  
 </item>  
 </argument>  
</actionsColumn>

- Tạo thư mục **Ui\Component\Listing\Column**và class tương ứng **Manager\Banner\Ui\Component\Listing\Column\Actions**với nội dung như sau (tham khảo module *Cms*, class *PageActions*):

<?php  
  
**namespace** Manager\Banner\Ui\Component\Listing\Column;  
  
**use** Magento\Framework\View\Element\UiComponent\ContextInterface;  
**use** Magento\Framework\View\Element\UiComponentFactory;  
**use** Magento\Ui\Component\Listing\Columns\Column;  
**use** Magento\Cms\Block\Adminhtml\Page\Grid\Renderer\Action\UrlBuilder;  
**use** Magento\Framework\UrlInterface;  
  
**class** Actions **extends** Column  
{  
 **const** BANNER\_URL\_PATH\_EDIT = 'banner/index/edit';  
 **const** BANNER\_URL\_PATH\_DELETE = 'banner/index/delete';  
  
 **protected** $actionUrlBuilder;  
 **protected** $urlBuilder;  
  
 **private** $editUrl;  
  
 **public function** \_\_construct(  
 ContextInterface $context,  
 UiComponentFactory $uiComponentFactory,  
 UrlBuilder $actionUrlBuilder,  
 UrlInterface $urlBuilder,  
 **array** $components = [],  
 **array** $data = [],  
 $editUrl = self::BANNER\_URL\_PATH\_EDIT  
 ) {  
 $this->urlBuilder = $urlBuilder;  
 $this->actionUrlBuilder = $actionUrlBuilder;  
 $this->editUrl = $editUrl;  
 parent::\_\_construct($context, $uiComponentFactory, $components, $data;  
 }  
  
 **public function** prepareDataSource(**array** $dataSource)  
 {  
 **if** (**isset**($dataSource['data']['items'])) {  
 // Lấy ra tên cột hiển thị (ở đây là cột actions)  
 $fieldName = $this->getData('name');  
  
 // Sửa lại giá trị của data source  
 **foreach** ($dataSource['data']['items'] **as** & $item) {  
 **if** (**isset**($item['id'])) {  
 // Thêm link đến trang Edit  
 $item[$fieldName]['edit'] = [  
 'href' => $this->urlBuilder->getUrl($this->editUrl, ['id' => $item['id']]),  
 'label' => \_\_('Edit')  
 ];  
  
 // Thêm link đến trang Delete  
 $item[$fieldName]['delete'] = [  
 'href' => $this->urlBuilder->getUrl(self::BANNER\_URL\_PATH\_DELETE, ['id' => $item['id']]),  
 'label' => \_\_('Delete'),  
 'confirm' => [  
 'title' => \_\_('Delete ${ $.$data.image }'),  
 'message' => \_\_('Are you sure you wan\'t to delete this record?')  
 ]  
 ];  
 }  
 }  
 }  
 **return** $dataSource;  
 }  
}

* Xử lý action Controller/Adminhtml/Index/Save.php
* Dùng để lưu database cho nut save, saveancontinue,

## 8 - Image Uploader

Sửa form-ui phần file image

Chú ý sửa tag

<item name="formElement" xsi:type="string">fileUploader</item>

Dể up load dc ảnh

<item name="url" xsi:type="url" path="banner/index\_image/upload"/>

Để lưu ảnh

<field name="images">  
 <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">banner</item>  
 <item name="label" xsi:type="string" translate="true"> Banner Image</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">true</item>  
 <item name="uploaderConfig" xsi:type="array">  
 <item name="url" xsi:type="url" path="banner/index\_image/upload"/>  
 </item>  
 </item>  
 </argument>  
</field>

* Tạo file Controller/Adminhtml/index/Image/Upload.php

**public function** execute()  
{  
 // Save image to temp folder  
 **try** {  
 $result = $this->imageUploader->saveFileToTmpDir('images');  
 $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);  
}

them đoạn mã vào /etc/di.xml tạo các class cho việc upload ảnh (index class vào contructor để xly ảnh)

<type name="Manager\Banner\Controller\Adminhtml\Index\Image\Upload">  
 <arguments>  
 <argument name="imageUploader" xsi:type="object">Manager\Banner\BannerImageUpload</argument>  
 </arguments>  
</type>  
  
<virtualType name="Manager\Banner\BannerImageUpload" type="Magento\Catalog\Model\ImageUploader">  
 <arguments>  
 <argument name="baseTmpPath" xsi:type="string">banner/tmp/images</argument>  
 <argument name="basePath" xsi:type="string">banner/images</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>

Tại file này cần chú ý đến

* basePath : thư mục tạm để chứa ảnh
* basePath : đường dẫn thực để chứa ảnh khi ng dùng nhấn save thì cop ảnh từ thư mục basePath vào basePath và lưu vào database

sửa action Save.php (Controller) them đoạn code xử lý ảnh

xly ảnh  
 **if** (**empty**($data['images'])) {  
 $data['images'] = **null**;  
 } **else** {  
 **if** ($data['images'][0] && $data['images'][0]['name'])  
 $data['image'] = $data['images'][0]['name'];  
 **else** $data['image'] = **null**;  
 }

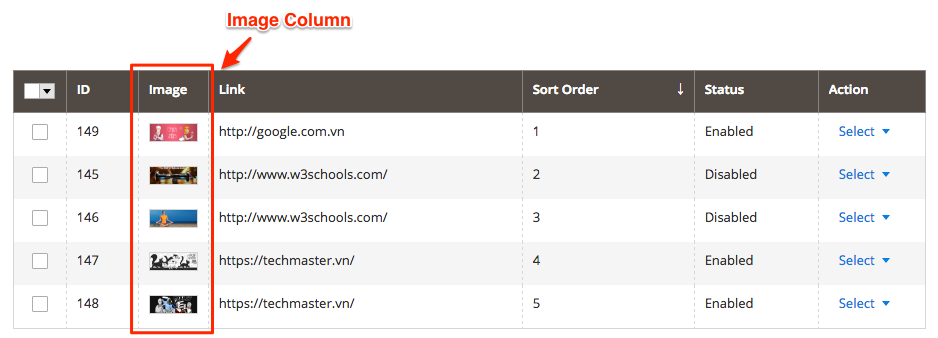
dataprovider trong model/banner

**<?php  
namespace** Manager\Banner\Model\Banner;  
**use** Manager\Banner\Model\ResourceModel\Banner\CollectionFactory;  
**use** Magento\Framework\App\Request\DataPersistorInterface;  
**class** DataProvider **extends** \Magento\Ui\DataProvider\AbstractDataProvider  
{  
 **protected** $collection;  
 **protected** $dataPersistor;  
 **protected** $loadedData;  
 **protected** $storeManager;  
 **public function** \_\_construct(  
 $name,  
 $primaryFieldName,  
 $requestFieldName,  
 CollectionFactory $bannerCollectionFactory,  
 DataPersistorInterface $dataPersistor,  
 \Magento\Store\Model\StoreManagerInterface $storeManager,  
 **array** $meta = [],  
 **array** $data = []  
 )  
 {  
 $this->collection = $bannerCollectionFactory->create();  
 $this->dataPersistor = $dataPersistor;  
 $this->storeManager = $storeManager;  
 **parent**::*\_\_construct*($name, $primaryFieldName, $requestFieldName, $meta, $data);  
 $this->meta = $this->prepareMeta($this->meta);  
 }  
 */\*\*  
 \* Prepares Meta  
 \*/* **public function** prepareMeta(**array** $meta)  
 {  
 **return** $meta;  
 }  
 */\*\*  
 \* Get data  
 \*/* **public function** getData()  
 {  
 // Get items  
 **if** (**isset**($this->loadedData)) {  
 **return** $this->loadedData;  
 }  
 $items = $this->collection->getItems();  
  
 **foreach** ($items **as** $banner) {  
 $data = $banner->getData();  
 $image = $data['image'];  
 **if** ($image && is\_string($image)) {  
 $data['images'][0]['name'] = $image;  
 $data['images'][0]['url'] = $this->storeManager->getStore()  
 ->getBaseUrl(\Magento\Framework\UrlInterface::*URL\_TYPE\_MEDIA*)  
 . 'banner/images/' . $image;  
 }  
 $this->loadedData[$banner->getId()] = $data;  
 }  
 $data = $this->dataPersistor->get('banner');  
 **if** (!**empty**($data)) {  
 $banner = $this->collection->getNewEmptyItem();  
 $banner->setData($data);  
 $this->loadedData[$banner->getId()] = $banner->getData();  
 $this->dataPersistor->clear('banner');  
 }  
 **return** $this->loadedData;  
 }  
}

Trong resource model thêm

**protected function** \_afterSave(\Magento\Framework\Model\AbstractModel $object)  
{  
 // Get image data before and after save  
 $oldImage = $object->getOrigData('image');  
 $newImage = $object->getData('image');  
 // Check when new image uploaded  
 **if** ($newImage != **null** && $newImage != $oldImage) {  
 $imageUploader = \Magento\Framework\App\ObjectManager::*getInstance*()  
 ->get('Manager\Banner\BannerImageUpload');  
 $imageUploader->moveFileFromTmp($newImage);  
 }  
 **return** $this;  
}

**Hướng dẫn hiển thị ảnh ngoài trang List**

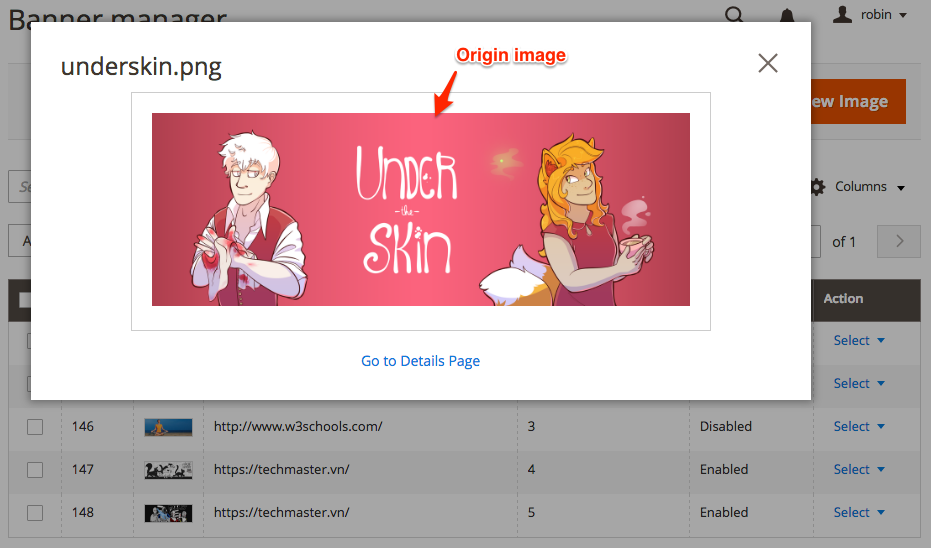


- Cấu hình class xử lý logic để tạo dữ liệu cho cột **image**(file banner\_listing.xml):

<*column name*="image" **class**="Manager\Banner\Ui\Component\Listing\Columns\Thumbnail">  
 <*argument name*="data" xsi:*type*="array">  
 <*item name*="config" xsi:*type*="array">  
 <*item name*="component" xsi:*type*="string">*Magento\_Ui*/*js*/*grid*/*columns*/*thumbnail*</*item*>  
 <*item name*="sortable" xsi:*type*="boolean">**false**</*item*>  
 <*item name*="altField" xsi:*type*="string">*link*</*item*>  
 <*item name*="has\_preview" xsi:*type*="string">1</*item*>  
 <*item name*="label" xsi:*type*="string" *translate*="true">*Image*</*item*>  
 </*item*>  
 </*argument*>  
</*column*>

- Tạo class tương ứng **Manager\Banner\Ui\Component\Listing\Column\Thumbnail**với nội dung như sau (tham khảo module *Catalog,*class*Thumbnail*):

<?*php***namespace** Manager\Banner\Ui\Component\Listing\Columns;  
**use** Magento\Framework\View\Element\UiComponentFactory;  
**use** Magento\Framework\View\Element\UiComponent\ContextInterface;  
  
**class** Thumbnail **extends** \Magento\Ui\Component\Listing\Columns\Column  
{  
 **protected** $storeManager;  
 **public function** \_\_construct(  
 ContextInterface $context,  
 UiComponentFactory $uiComponentFactory,  
 \Magento\Store\Model\StoreManagerInterface $storeManager,  
 \Magento\Framework\UrlInterface $urlBuilder,  
 **array** $components = [],  
 **array** $data = []  
 )  
 {  
 **parent**::*\_\_construct*($context, $uiComponentFactory, $components, $data);  
 $this->storeManager = $storeManager;  
 $this->urlBuilder = $urlBuilder;  
 }  
 **public function** prepareDataSource(**array** $dataSource)  
 {  
 **if** (**isset**($dataSource['data']['items'])) {  
 // Lấy ra tên cột hiển thị (ở đây là cột image)  
 $fieldName = $this->getData('name');  
  
 // Sửa lại giá trị của data source  
 **foreach** ($dataSource['data']['items'] **as** & $item) {  
 $url = '';  
 **if** ($item[$fieldName] != '') {  
 // Lấy đường dẫn của ảnh  
 $url = $this->storeManager->getStore()->getBaseUrl(  
 \Magento\Framework\UrlInterface::*URL\_TYPE\_MEDIA* ) . 'banner/images/' . $item[$fieldName];  
 }  
  
 /\*   
 \* Truyền vào các giá trị cần thiết   
 \* $fieldName . '\_src' - Đường dẫn của ảnh trong bảng  
 \* $fieldName . '\_alt' - Giá trị thuộc tính alt của ảnh  
 \* $fieldName . '\_link' - Link khi bấm vào ảnh (trỏ sang trang edit)  
 \* $fieldName . '\_orig\_src' - Đường dẫn ảnh khi phóng to  
 \*/  
 $item[$fieldName . '\_src'] = $url;  
 $item[$fieldName . '\_alt'] = $item[$fieldName];  
 $item[$fieldName . '\_link'] = $this->urlBuilder->getUrl(  
 'banner/index/edit',  
 ['id' => $item['id']]  
 );  
 $item[$fieldName . '\_orig\_src'] = $url;  
 }  
 }  
 **return** $dataSource;  
 }  
}



## 9 - Inline edit, Mass action

**InlineEdit**

* giống action Edit nhưng inlineEdit chỉnh sửa trực tiếp ở list và dùng aijax gửi lên sever gửi lại data dạng json
* ở file listing phần <colums>

<settings>  
 <editorConfig>  
 <param name="clientConfig" xsi:type="array">  
 <item name="saveUrl" xsi:type="url" path="manager/banner/inlineEdit"/>  
 <item name="validateBeforeSave" xsi:type="boolean">false</item>  
 </param>  
 <param name="indexField" xsi:type="string">id</param>  
 <param name="enabled" xsi:type="boolean">true</param>  
 <param name="selectProvider" xsi:type="string">manager\_banner\_listing.manager\_banner\_listing.manager\_banner\_columns.ids</param>  
 </editorConfig>  
 <childDefaults>  
 <param name="fieldAction" xsi:type="array">  
 <item name="provider" xsi:type="string">manager\_banner\_listing.manager\_banner\_listing.manager\_banner\_columns\_editor</item>  
 <item name="target" xsi:type="string">startEdit</item>  
 <item name="params" xsi:type="array">  
 <item name="0" xsi:type="string">${ $.$data.rowIndex }</item>  
 <item name="1" xsi:type="boolean">true</item>  
 </item>  
 </param>  
 </childDefaults>  
</settings>

* editorConfig dùng để tạo chức năng inlineEdit
* saveurl có path gọi đến inlineEdit (đây code xly phần inline)
* note (các field sd inlinedit)
  + listing ui các colums có validate name khác nhau
  + action cần chọn đúng đường dẫn

**Mass Delete**

- Sửa cấu hình **massaction** trong **banner\_listing.xml** cho action delete, url trỏ đến **banner/index/massDelete**:

<*massaction name*="listing\_massaction">  
 <*action name*="delete">  
 <*argument name*="data" xsi:*type*="array">  
 <*item name*="config" xsi:*type*="array">  
 <*item name*="type" xsi:*type*="string">*delete*</*item*>  
 <*item name*="label" xsi:*type*="string" *translate*="true">*Delete*</*item*>  
 <*item name*="url" xsi:*type*="url" *path*="banner/index/massDelete"/>  
 <*item name*="confirm" xsi:*type*="array">  
 <*item name*="title" xsi:*type*="string" *translate*="true">*Delete items*</*item*>  
 <*item name*="message" xsi:*type*="string" *translate*="true">*Are you sure you wan*'t to delete selected items?</item>  
 </item>  
 </item>  
 </argument>  
 </action>  
</massaction>

- Tạo action **MassDelete** (tham khảo module *Cms*):

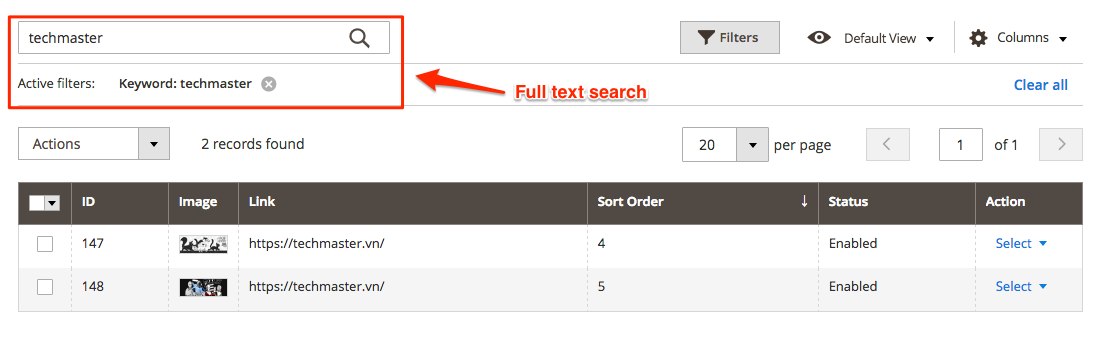
<?*php***namespace** Manager\Banner\Controller\Adminhtml\Index;  
**use** Magento\Framework\Controller\ResultFactory;  
**use** Magento\Backend\App\Action\Context;  
**use** Magento\Ui\Component\MassAction\Filter;  
**use** Manager\Banner\Model\ResourceModel\Banner\CollectionFactory;  
**class** MassDelete **extends** \Magento\Backend\App\Action  
{  
 **protected** $filter;  
 **protected** $collectionFactory;  
 **public function** \_\_construct(Context $context, Filter $filter, CollectionFactory $collectionFactory)  
 {  
 $this->filter = $filter;  
 $this->collectionFactory = $collectionFactory;  
 **parent**::*\_\_construct*($context);  
 }  
 **public function** execute()  
 {  
 // Lấy ra các bản ghi đang chọn sử dụng filter (Magento\Ui\Component\MassAction\Filter)  
 $collection = $this->filter->getCollection($this->collectionFactory->create());  
 // Đếm số bản ghi đang chọn  
 $collectionSize = $collection->getSize();  
 // Xóa bản ghi đang chọn  
 **foreach** ($collection **as** $banner) {  
 $banner->delete();  
 }  
 // Thêm thông báo xóa thành công  
 $this->messageManager->addSuccess(\_\_('A total of %1 record(s) have been deleted.', $collectionSize));  
 // Redirect lại về trang List  
 $resultRedirect = $this->resultFactory->create(ResultFactory::*TYPE\_REDIRECT*);  
 **return** $resultRedirect->setPath('\*/\*/');  
 }  
}

- Bổ sung thêm (ghi đè) thuộc tính **\_idFieldName** vào Collection của Model

<?*php***namespace** Manager\Banner\Model\ResourceModel\Banner;  
**class** Collection **extends** \Magento\Framework\Model\ResourceModel\Db\Collection\AbstractCollection  
{  
 **protected** $\_idFieldName = 'id';  
 **protected function** \_construct()  
 {  
 // Model + Resource Model  
 $this->\_init('Manager\Banner\Model\Banner', 'Manager\Banner\Model\ResourceModel\Banner');  
 }  
}

- Các **massaction** khác như **MassEnable**, **MassDisable** tương tự.

**Hướng dẫn chức năng full text search**

****

- Nâng cấp module (tăng version), bổ sung thêm index vào bảng **banner**bằng file**UpgradeSchema.php**

**public function** upgrade(SchemaSetupInterface $setup, ModuleContextInterface $context)  
{  
 $setup->startSetup();  
 $connection = $setup->getConnection();  
 $tableName = $setup->getTable('banner');  
 // Tạo index kiểu FULLTEXT trên 2 cột là image và link  
 $connection->addIndex(  
 $tableName,  
 'search',  
 [  
 'image',  
 'link'  
 ],  
 \Magento\Framework\DB\Adapter\AdapterInterface::*INDEX\_TYPE\_FULLTEXT* );  
 $setup->endSetup();  
}

- Update lại module bằng lệnh **sudo bin/magento setup:upgrade**

Note :

* namespace ()

## 10 - Widget

Cấu hình hiển thị ngoài frontend

* tạo file /etc/widget.xml
  + id
  + clsass(la 1 block)

Manager\Banner\Block\WidgetBanner

# Tab Form (Add Products Grid to Custom Admin Module)

Tạo module

Tạo database ()

Tạo dc list

Tạo form add bang mẫu tabform

Để tạo được tab phải thay đổi layout xml của page cần chỉnh sửa

…\view\adminhtml\layout\wsproductsgrid\_contacts\_edit.xml

<?xml version="1.0"?>  
<page xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" layout="admin-2columns-left" xsi:noNamespaceSchemaLocation="urn:magento:framework:View/Layout/etc/page\_configuration.xsd">  
<update handle="editor"/>  
<body>

<!—xử lý các nút bên page edit -->  
 <referenceContainer name="content">  
 <block class="Webspeaks\ProductsGrid\Block\Adminhtml\Contact\Edit" name="wsproductsgrid\_contact\_edit"/>  
 </referenceContainer>

<!— content cho các tab và noi dung của tab-->

<referenceContainer name="left">  
 <block class="Webspeaks\ProductsGrid\Block\Adminhtml\Contact\Edit\Tabs" name="wsproductsgrid\_contact\_edit\_tabs">  
 <block class="Webspeaks\ProductsGrid\Block\Adminhtml\Contact\Edit\Tab\Main" name="wsproductsgrid\_contact\_edit\_tab\_main"/>

<!—tạo ra các nội dung tab main -->  
 <action method="addTab">  
 <argument name="name" xsi:type="string">main\_section</argument>  
 <argument name="block" xsi:type="string">wsproductsgrid\_contact\_edit\_tab\_main</argument>  
 </action>  
 <!-- This code is added for products grid -->

<!—tạo ra các nội dung tab product -->  
 <action method="addTab">  
 <argument name="name" xsi:type="string">attachment\_products</argument>  
 <argument name="block" xsi:type="array">  
 <item name="label" xsi:type="string">Select Products</item>  
 <item name="title" xsi:type="string">Select Products</item>  
 <item name="url" xsi:type="helper" helper="Webspeaks\ProductsGrid\Helper\Data::getProductsGridUrl"></item>  
 <item name="class" xsi:type="string">ajax</item>  
 </argument>  
 </action>  
 <!-- This code is added for products grid -->  
 </block>  
 </referenceContainer>  
</body>  
</page>

Tạo helper

app\code\Webspeaks\ProductsGrid\Helper\Data.php

**<?php  
  
namespace** Webspeaks\ProductsGrid\Helper;  
  
**class** Data **extends** \Magento\Framework\App\Helper\AbstractHelper  
{  
  
 */\*\*  
 \** ***@var*** *\Magento\Backend\Model\UrlInterface  
 \*/* **protected** $\_backendUrl;  
  
 */\*\*  
 \** ***@var*** *\Magento\Store\Model\StoreManagerInterface $storeManager  
 \*/* **protected** $storeManager;  
  
 */\*\*  
 \** ***@param*** *\Magento\Framework\App\Helper\Context $context  
 \** ***@param*** *\Magento\Backend\Model\UrlInterface $backendUrl  
 \*/* **public function** \_\_construct(  
 \Magento\Framework\App\Helper\Context $context,  
 \Magento\Backend\Model\UrlInterface $backendUrl,  
 \Magento\Store\Model\StoreManagerInterface $storeManager  
 ) {  
 **parent**::*\_\_construct*($context);  
 $this->\_backendUrl = $backendUrl;  
 $this->storeManager = $storeManager;  
 }  
  
 */\*\*  
 \* get products tab Url in admin  
 \** ***@return*** *string  
 \*/* **public function** getProductsGridUrl()  
 {  
 **return** $this->\_backendUrl->getUrl('wsproductsgrid/contacts/products', ['\_current' => **true**]);  
 }  
}

# Grid admin

## Listing

### Tạo db

### Tao module

### Tạo model (model, resourcemodel, collection)

### routes admin

…/etc/adminhtml/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 id="grid" frontName="grid">  
 <module name="Manager\_Grid"/>  
 </route>  
 </router>  
</config>

### **admin menu**

…/etc/adminhtml/menu.xml

Tạo menu cho action của module

<?xml version="1.0"?>  
  
<config xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="urn:magento:module:Magento\_Backend:etc/menu.xsd">  
 <menu>  
 <add id="Manager\_Grid" title="Grid Banner" translate="title" module="Manager\_Grid" sortOrder="60"  
 parent="Manager\_Banner::manager" action="grid/banner" resource="Manager\_Banner::banner"/>  
 </menu>  
</config>

Chú ý

* Action phải chỉ đúng controller của action đó
* Id phải duy nhất
* Title
* Parent
* Resource
* Module
* SortOrder

### **Controller**

Controller/Adminhtml/Banner/Index.php

**<?php  
  
namespace** Manager\Grid\Controller\Adminhtml\Banner;  
  
**class** Index **extends** \Magento\Backend\App\Action  
{  
 **protected** $resultPageFactory;  
  
 **public function** \_\_construct(  
 \Magento\Backend\App\Action\Context $context,  
 \Magento\Framework\View\Result\PageFactory $resultPageFactory  
 )  
 {  
 **parent**::*\_\_construct*($context);  
 $this->resultPageFactory = $resultPageFactory;  
 }  
  
 **public function** execute()  
 {

// trả về page   
 $resultPage = $this->resultPageFactory->create();  
 $resultPage->getConfig()->getTitle()->prepend((\_\_('Grids Banner')));  
  
 **return** $resultPage;  
 }  
  
}

#### **Create block for this grid**

…/Block/Adminhtml/Banner.php

Khai báo những thông tin cần thiết cho layout

* Controller là đường dẫn đến khối Lưới bên trong thư mục Block (…/controller/adminhtml/banner)
* blockGroup là tên của mô-đun
* text : tiêu đề trang
* button (addButtonLable) nhãn nút thêm

**<?php  
namespace** Manager\Grid\Block\Adminhtml;  
  
**class** Banner **extends** \Magento\Backend\Block\Widget\Grid\Container  
{  
  
 **protected function** \_construct()  
 {  
 $this->\_objectId = 'id';  
// $this->\_objectId = 'contact\_id';  
 $this->\_controller = 'adminhtml\_banner';  
 $this->\_blockGroup = 'Manager\_Grid';  
 $this->\_headerText = \_\_('Banners');  
 $this->\_addButtonLabel = \_\_('Create New Banner');  
 **parent**::*\_construct*();  
 }  
}

#### **Create layout file**

Nội dung hiển thỉ danh sách tất cả các bản ghi lấy ra từ db

view/adminhtml/layout/grid\_banner\_index.xml

chú ý tên file trên ( path/controller/action)

<?xml version="1.0"?>  
<page xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="urn:magento:framework:View/Layout/etc/page\_configuration.xsd">  
 <update handle="styles"/>  
 <body>  
 <referenceContainer name="content">  
 <block class="Manager\Grid\Block\Adminhtml\Banner" name="grid\_banner\_index">  
 <block class="Magento\Backend\Block\Widget\Grid" name="grid\_banner\_index.grid" as="grid">  
 <arguments>  
 <argument name="id" xsi:type="string">id</argument>  
 <argument name="dataSource" xsi:type="object">Manager\Grid\Model\ResourceModel\Banner\Collection</argument>  
 <argument name="default\_sort" xsi:type="string">id</argument>  
 <argument name="default\_dir" xsi:type="string">ASC</argument>  
 <argument name="save\_parameters\_in\_session" xsi:type="string">1</argument>  
 </arguments>  
 <block class="Magento\Backend\Block\Widget\Grid\ColumnSet" name="grid\_banner\_index.grid.columnSet" as="grid.columnSet">  
 <arguments>  
 <argument name="rowUrl" xsi:type="array">  
 <item name="path" xsi:type="string">\*/\*/edit</item>  
 </argument>  
 </arguments>  
 <block class="Magento\Backend\Block\Widget\Grid\Column" as="id">  
 <arguments>  
 <argument name="header" xsi:type="string" translate="true">ID</argument>  
 <argument name="index" xsi:type="string">id</argument>  
 <argument name="type" xsi:type="string">text</argument>  
 <argument name="column\_css\_class" xsi:type="string">col-id</argument>  
 <argument name="header\_css\_class" xsi:type="string">col-id</argument>  
 </arguments>  
 </block>  
 <block class="Magento\Backend\Block\Widget\Grid\Column" as="title">  
 <arguments>  
 <argument name="header" xsi:type="string" translate="true">Title</argument>  
 <argument name="index" xsi:type="string">title</argument>  
 <argument name="type" xsi:type="string">text</argument>  
 <argument name="column\_css\_class" xsi:type="string">title</argument>  
 <argument name="header\_css\_class" xsi:type="string">title</argument>  
 </arguments>  
 </block>  
 <!--<block class="Magento\Backend\Block\Widget\Grid\Column" as="image">-->  
 <!--<arguments>-->  
 <!--<argument name="header" xsi:type="string" translate="true">Image</argument>-->  
 <!--<argument name="index" xsi:type="string">image</argument>-->  
 <!--<argument name="type" xsi:type="string">text</argument>-->  
 <!--<argument name="column\_css\_class" xsi:type="string">image</argument>-->  
 <!--<argument name="header\_css\_class" xsi:type="string">image</argument>-->  
 <!--</arguments>-->  
 <!--</block>-->  
 <block class="Magento\Backend\Block\Widget\Grid\Column" as="link">  
 <arguments>  
 <argument name="header" xsi:type="string" translate="true">Link</argument>  
 <argument name="index" xsi:type="string">link</argument>  
 <argument name="type" xsi:type="string">text</argument>  
 <argument name="column\_css\_class" xsi:type="string">link</argument>  
 <argument name="header\_css\_class" xsi:type="string">link</argument>  
 </arguments>  
 </block>  
 <block class="Magento\Backend\Block\Widget\Grid\Column" as="created\_at">  
 <arguments>  
 <argument name="header" xsi:type="string" translate="true">Created</argument>  
 <argument name="index" xsi:type="string">created\_at</argument>  
 <argument name="type" xsi:type="string">date</argument>  
 <argument name="column\_css\_class" xsi:type="string">created\_at</argument>  
 <argument name="header\_css\_class" xsi:type="string">created\_at</argument>  
 </arguments>  
 </block>  
 <block class="Magento\Backend\Block\Widget\Grid\Column" as="updated\_at">  
 <arguments>  
 <argument name="header" xsi:type="string" translate="true">Modified</argument>  
 <argument name="index" xsi:type="string">updated\_at</argument>  
 <argument name="type" xsi:type="string">date</argument>  
 <argument name="column\_css\_class" xsi:type="string">updated\_at</argument>  
 <argument name="header\_css\_class" xsi:type="string">updated\_at</argument>  
 </arguments>  
 </block>  
  
 </block>  
 <block class="Magento\Backend\Block\Widget\Grid\Massaction" name="manager.grid.massaction" as="grid.massaction">  
 <arguments>  
 <argument name="massaction\_id\_field" xsi:type="string">id</argument>  
 <argument name="form\_field\_name" xsi:type="string">ids</argument>  
 <argument name="use\_select\_all" xsi:type="string">1</argument>  
 <argument name="options" xsi:type="array">  
 <item name="delete" xsi:type="array">  
 <item name="label" xsi:type="string" translate="true">Delete</item>  
 <item name="url" xsi:type="string">\*/\*/massDelete</item>  
 </item>  
 <item name="enable" xsi:type="array">  
 <item name="label" xsi:type="string" translate="true">Enable</item>  
 <item name="url" xsi:type="string">\*/\*/massEnable</item>  
 </item>  
 </argument>  
 </arguments>  
 </block>  
  
 </block>  
 </block>  
 </referenceContainer>  
 </body>  
</page>

ở đây chú ý tên trường của bảng trong db

## New and Edit

Tạo action cho 2 chức năng vì chung form

* tạo controller

…/Controller/Adminhtml/Banner/NewAction.php

**<?php  
  
namespace** Manager\Grid\Controller\Adminhtml\Banner;  
  
**class** NewAction **extends** \Magento\Backend\App\Action  
{  
 **protected** $resultForwardFactory;**public function** \_\_construct(  
 \Magento\Backend\App\Action\Context $context,  
 \Magento\Backend\Model\View\Result\ForwardFactory $resultForwardFactory  
 ) {  
 $this->resultForwardFactory = $resultForwardFactory;  
 **parent**::*\_\_construct*($context);  
 }**protected function** \_isAllowed()  
 {  
 **return true**;  
 }  
**public function** execute()  
 {  
 */\*\** ***@var*** *\Magento\Backend\Model\View\Result\Forward $resultForward \*/* $resultForward = $this->resultForwardFactory->create();  
 **return** $resultForward->forward('edit');  
 }  
}

…/Controller/Adminhtml/Banner/Edit.php

**<?php  
  
namespace** Manager\Grid\Controller\Adminhtml\Banner;  
  
**use** Magento\Backend\App\Action;  
  
**class** Edit **extends** \Magento\Backend\App\Action  
{

**protected** $\_coreRegistry = **null**;  
  
 **protected** $resultPageFactory;  
**public function** \_\_construct(  
 Action\Context $context,  
 \Magento\Framework\View\Result\PageFactory $resultPageFactory,  
 \Magento\Framework\Registry $registry  
 ) {  
 $this->resultPageFactory = $resultPageFactory;  
 $this->\_coreRegistry = $registry;  
 **parent**::*\_\_construct*($context);  
 }  
**protected function** \_initAction()  
 {  
 $resultPage = $this->resultPageFactory->create();  
 **return** $resultPage;  
 }  
**public function** execute()  
 {  
 $id = $this->getRequest()->getParam(id');  
 $model = $this->\_objectManager->create('Manager\Grid\Model\Banner);  
  
 **if** ($id) {  
 $model->load($id);  
 **if** (!$model->getId()) {  
 $this->messageManager->addError(\_\_('This contact no longer exists.'));  
$resultRedirect = $this->resultRedirectFactory->create();  
  
 **return** $resultRedirect->setPath('\*/\*/');  
 }  
 }  
 $data = $this->\_objectManager->get('Magento\Backend\Model\Session')->getFormData(**true**);  
 **if** (!**empty**($data)) {  
 $model->setData($data);  
 }  
 $this->\_coreRegistry->register('mn\_banner', $model);  
  
 */\*\** ***@var*** *\Magento\Backend\Model\View\Result\Page $resultPage \*/* $resultPage = $this->\_initAction();  
 $resultPage->addBreadcrumb(  
 $id ? \_\_('Edit Contact') : \_\_('New Contact'),  
 $id ? \_\_('Edit Contact') : \_\_('New Contact')  
 );  
 $resultPage->getConfig()->getTitle()->prepend(\_\_('Banner'));  
 $resultPage->getConfig()->getTitle()  
 ->prepend($model->getId() ? $model->getTitle() : \_\_('New Contact'));  
  
 **return** $resultPage;  
 }  
}

tạo block để tạo ra khung của from (có thể tạo thêm tab form) ở đây tạo tab form, còn k thì như index gọi đến layout luôn

* tạo …/Block/Adminhtml/Banner/Banner.php

tạo ra các nút như save, saveandcontinue delete, back

**<?php  
  
namespace** Manager\Grid\Block\Adminhtml\Banner;  
  
**class** Edit **extends** \Magento\Backend\Block\Widget\Form\Container  
{  
  
 **protected** $\_coreRegistry = **null**;  
  
 **public function** \_\_construct(  
 \Magento\Backend\Block\Widget\Context $context,  
 \Magento\Framework\Registry $registry,  
 **array** $data = []  
 ) {  
 $this->\_coreRegistry = $registry;  
 **parent**::*\_\_construct*($context, $data);  
 }  
 **protected function** \_construct()  
 {  
 $this->\_objectId = 'id';  
 $this->\_blockGroup = 'Manager\_Grid';  
 $this->\_controller = 'adminhtml\_banner';  
 **parent**::*\_construct*();  
 $this->buttonList->update('save', 'label', \_\_('Save Banner'));  
 $this->buttonList->add(  
 'saveandcontinue',  
 [  
 'label' => \_\_('Save and Continue Edit'),  
 'class' => 'save',  
 'data\_attribute' => [  
 'mage-init' => [  
 'button' => ['event' => 'saveAndContinueEdit', 'target' => '#edit\_form'],  
 ],  
 ]  
 ],  
 -100  
 );  
  
 $this->buttonList->update('delete', 'label', \_\_('Delete Banner'));  
 }  
**protected function** \_isAllowedAction($resourceId)  
 {  
 **return** $this->\_authorization->isAllowed($resourceId);  
 }  
  
 **protected function** \_getSaveAndContinueUrl()  
 {  
 **return** $this->getUrl('grid/\*/save', ['\_current' => **true**, 'back' => 'edit', 'active\_tab' => '']);  
 }  
  
}

Tạo khung chứa tab cho form

**<?php  
  
namespace** Manager\Grid\Block\Adminhtml\Banner\Edit;  
**class** Tabs **extends** \Magento\Backend\Block\Widget\Tabs  
{  
 **protected function** \_construct()  
 {  
 **parent**::*\_construct*();  
 $this->setId('banner\_tabs');  
 $this->setDestElementId('edit\_form');  
 $this->setTitle(\_\_('Banner Information'));  
 }  
}

Tạo nội dung cho form ở tab đó

**<?php  
  
namespace** Manager\Grid\Block\Adminhtml\Banner\Edit;  
**class** Form **extends** \Magento\Backend\Block\Widget\Form\Generic  
{  
**protected function** \_prepareForm()  
 {  
 */\*\** ***@var*** *\Magento\Framework\Data\Form $form \*/* $form = $this->\_formFactory->create(  
 ['data' => ['id' => 'edit\_form', 'action' => $this->getData('action'), 'method' => 'post']]  
 );  
 $form->setUseContainer(**true**);  
 $this->setForm($form);  
 **return parent**::*\_prepareForm*();  
 }  
}

thêm phần tử cho form

**<?php  
namespace** Manager\Grid\Block\Adminhtml\Banner\Edit\Tab;  
  
**class** Main **extends** \Magento\Backend\Block\Widget\Form\Generic **implements** \Magento\Backend\Block\Widget\Tab\TabInterface  
{**protected** $store;  
**protected** $helper;  
**public function** \_\_construct(  
 \Magento\Backend\Block\Template\Context $context,  
 \Magento\Framework\Registry $registry,  
 \Magento\Framework\Data\FormFactory $formFactory,  
 \Manager\Grid\Helper\Data $helper,  
 **array** $data = []  
 ) {  
 $this->helper = $helper;  
 **parent**::*\_\_construct*($context, $registry, $formFactory, $data);  
 }  
**protected function** \_prepareForm()  
 {  
 */\** ***@var*** *$model \Manager\Grid\Model\Contact \*/* $model = $this->\_coreRegistry->registry('mn\_banner');  
 */\*\** ***@var*** *\Magento\Framework\Data\Form $form \*/* $form = $this->\_formFactory->create();  
  
 $form->setHtmlIdPrefix('banner\_');  
  
 $fieldset = $form->addFieldset('base\_fieldset', ['legend' => \_\_('Banner Information')]);  
  
 **if** ($model->getId()) {  
 $fieldset->addField('id', 'hidden', ['name' => 'id']);  
 }  
  
 $fieldset->addField(  
 'contact\_name',  
 'text',  
 [  
 'name' => 'contact\_name',  
 'label' => \_\_('Name'),  
 'title' => \_\_('Name'),  
 'required' => **true**,  
 ]  
 );  
  
 $fieldset->addField(  
 'age',  
 'text',  
 [  
 'name' => 'age',  
 'label' => \_\_('Age'),  
 'title' => \_\_('Age'),  
 'required' => **true**,  
 ]  
 );  
  
 $fieldset->addField(  
 'phone',  
 'text',  
 [  
 'name' => 'phone',  
 'label' => \_\_('Phone'),  
 'title' => \_\_('Phone'),  
 'required' => **true**,  
 ]  
 );  
  
 $fieldset->addField(  
 'address',  
 'textarea',  
 [  
 'name' => 'address',  
 'label' => \_\_('Address'),  
 'title' => \_\_('Address'),  
 'required' => **true**,  
 ]  
 );  
  
 $form->setValues($model->getData());  
 $this->setForm($form);  
  
 **return parent**::*\_prepareForm*();  
 }  
**public function** getTabLabel()  
 {  
 **return** \_\_('Main');  
 }  
**public function** getTabTitle()  
 {  
 **return** \_\_('Main');  
 }  
**public function** canShowTab()  
 {  
 **return true**;  
 }  
**public function** isHidden()  
 {  
 **return false**;  
 }  
**protected function** \_isAllowedAction($resourceId)  
 {  
 **return** $this->\_authorization->isAllowed($resourceId);  
 }  
}

Noi dung cho tab thu 2

**<?php  
  
namespace** Manager\Grid\Block\Adminhtml\Banner\Edit\Tab;  
  
**use** Manager\Grid\Model\ContactFactory;  
  
**class** Products **extends** \Magento\Backend\Block\Widget\Grid\Extended  
{  
**protected** $productCollectionFactory;  
**protected** $contactFactory;  
**protected** $registry;  
  
 **protected** $\_objectManager = **null**;  
  
 **public function** \_\_construct(  
 \Magento\Backend\Block\Template\Context $context,  
 \Magento\Backend\Helper\Data $backendHelper,  
 \Magento\Framework\Registry $registry,  
 \Magento\Framework\ObjectManagerInterface $objectManager,  
 ContactFactory $contactFactory,  
 \Magento\Catalog\Model\ResourceModel\Product\CollectionFactory $productCollectionFactory,  
 **array** $data = []  
 ) {  
 $this->contactFactory = $contactFactory;  
 $this->productCollectionFactory = $productCollectionFactory;  
 $this->\_objectManager = $objectManager;  
 $this->registry = $registry;  
 **parent**::*\_\_construct*($context, $backendHelper, $data);  
 }  
**protected function** \_construct()  
 {  
 **parent**::*\_construct*();  
 $this->setId('productsGrid');  
 $this->setDefaultSort('entity\_id');  
 $this->setDefaultDir('DESC');  
 $this->setSaveParametersInSession(**true**);  
 $this->setUseAjax(**true**);  
 **if** ($this->getRequest()->getParam('id')) {  
 $this->setDefaultFilter(**array**('in\_product' => 1));  
 }  
 }  
**protected function** \_addColumnFilterToCollection($column)  
 {  
 **if** ($column->getId() == 'in\_product') {  
 $productIds = $this->\_getSelectedProducts();  
  
 **if** (**empty**($productIds)) {  
 $productIds = 0;  
 }  
 **if** ($column->getFilter()->getValue()) {  
 $this->getCollection()->addFieldToFilter('entity\_id', **array**('in' => $productIds));  
 } **else** {  
 **if** ($productIds) {  
 $this->getCollection()->addFieldToFilter('entity\_id', **array**('nin' => $productIds));  
 }  
 }  
 } **else** {  
 **parent**::*\_addColumnFilterToCollection*($column);  
 }  
  
 **return** $this;  
 }  
**protected function** \_prepareCollection()  
 {  
 $collection = $this->productCollectionFactory->create();  
 $collection->addAttributeToSelect('name');  
 $collection->addAttributeToSelect('sku');  
 $collection->addAttributeToSelect('price');  
 $this->setCollection($collection);  
 **return parent**::*\_prepareCollection*();  
 }  
  
 **protected function** \_prepareColumns()  
 {  
 */\** ***@var*** *$model \Manager\Grid\Model\Slide \*/* $model = $this->\_objectManager->get('\Manager\Grid\Model\Contact');  
  
 $this->addColumn(  
 'in\_product',  
 [  
 'header\_css\_class' => 'a-center',  
 'type' => 'checkbox',  
 'name' => 'in\_product',  
 'align' => 'center',  
 'index' => 'entity\_id',  
 'values' => $this->\_getSelectedProducts(),  
 ]  
 );  
  
 $this->addColumn(  
 'entity\_id',  
 [  
 'header' => \_\_('Product ID'),  
 'type' => 'number',  
 'index' => 'entity\_id',  
 'header\_css\_class' => 'col-id',  
 'column\_css\_class' => 'col-id',  
 ]  
 );  
 $this->addColumn(  
 'name',  
 [  
 'header' => \_\_('Name'),  
 'index' => 'name',  
 'class' => 'xxx',  
 'width' => '50px',  
 ]  
 );  
 $this->addColumn(  
 'sku',  
 [  
 'header' => \_\_('Sku'),  
 'index' => 'sku',  
 'class' => 'xxx',  
 'width' => '50px',  
 ]  
 );  
 $this->addColumn(  
 'price',  
 [  
 'header' => \_\_('Price'),  
 'type' => 'currency',  
 'index' => 'price',  
 'width' => '50px',  
 ]  
 );  
  
 **return parent**::*\_prepareColumns*();  
 }  
**public function** getGridUrl()  
 {  
 **return** $this->getUrl('\*/\*/productsgrid', ['\_current' => **true**]);  
 }  
**public function** getRowUrl($row)  
 {  
 **return** '';  
 }  
  
 **protected function** \_getSelectedProducts()  
 {  
 $contact = $this->getContact();  
 **return** $contact->getProducts($contact);  
 }  
  
 **public function** getSelectedProducts()  
 {  
 $contact = $this->getContact();  
 $selected = $contact->getProducts($contact);  
  
 **if** (!is\_array($selected)) {  
 $selected = [];  
 }  
 **return** $selected;  
 }  
  
 **protected function** getBanner()  
 {  
 $contactId = $this->getRequest()->getParam('id');  
 $contact = $this->contactFactory->create();  
 **if** ($contactId) {  
 $contact->load($contactId);  
 }  
 **return** $contact;  
 }  
**public function** canShowTab()  
 {  
 **return true**;  
 }

**public function** isHidden()  
 {  
 **return true**;  
 }  
}

Tạo layout cho page edit và new

new

<?xml version="1.0"?>  
<page xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" layout="admin-2columns-left" xsi:noNamespaceSchemaLocation="urn:magento:framework:View/Layout/etc/page\_configuration.xsd">  
 <update handle="grid\_banner\_edit"/>  
 <body/>  
</page>

Edit

<?xml version="1.0"?>  
<page xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" layout="admin-2columns-left" xsi:noNamespaceSchemaLocation="urn:magento:framework:View/Layout/etc/page\_configuration.xsd">  
 <update handle="editor"/>  
 <body>  
 <referenceContainer name="content">  
 <block class="Manager\Grid\Block\Adminhtml\Banner\Edit" name="grid\_banner\_edit"/>  
 </referenceContainer>  
 <referenceContainer name="left">  
 <block class="Manager\Grid\Block\Adminhtml\Banner\Edit\Tabs" name="grid\_banner\_edit\_tabs">  
 <block class="Manager\Grid\Block\Adminhtml\Banner\Edit\Tab\Main" name="grid\_banner\_edit\_tab\_main"/>  
 <action method="addTab">  
 <argument name="name" xsi:type="string">main\_section</argument>  
 <argument name="block" xsi:type="string">grid\_banner\_edit\_tab\_main</argument>  
 </action>  
 <action method="addTab">  
 <argument name="name" xsi:type="string">attachment\_products</argument>  
 <argument name="block" xsi:type="array">  
 <item name="label" xsi:type="string">Select Products</item>  
 <item name="title" xsi:type="string">Select Products</item>  
 <item name="url" xsi:type="helper" helper="Manager\Grid\Helper\Data::getGridUrl"></item>  
 <item name="class" xsi:type="string">ajax</item>  
 </argument>  
 </action>  
 </block>  
 </referenceContainer>  
 </body>  
</page>