# 1. Tạo module

# 2. Create New Tab in Admin Form

Tạo cho trang new hoặc edit

view\adminhtml\layout\[modulename]\_[controller]\_new.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="manager\_block\_edit"/>  
 <body/>  
</page>

Vì page new và edit chung form nên sử dụng như này sẽ ngắn hơn

Tạo tab form cần chỉnh sửa file xml trong layout của edit form

Chỉnh sửa layout (view\adminhtml\layout\[modulename]\_[controller]\_edit.xml)

### Create New Tab in Admin Form

To create the new tab, you have to modify the layout xml of form edit page. In our module this file is \[ModuleName]\view\adminhtml\layout\[modulename]\_[controller]\_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>  
 <referenceContainer name="content">  
 <block class="Webspeaks\ProductsGrid\Block\Adminhtml\Contact\Edit" name="wsproductsgrid\_contact\_edit"/>  
 </referenceContainer>  
 <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"/>  
 <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>  
 <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>  
 </block>  
 </referenceContainer>  
 </body>  
</page>

Hoặc có thể thêm tab ở trong [ModuleName]\Block\Adminhtml\Block\Edit\Tabs.php

**protected function** \_prepareLayout()  
{  
 $this->addTab(  
 'banner',  
 [  
 'label' => \_\_('Products'),  
 'url' => $this->getUrl('manager/\*/products', ['\_current' => **true**]),  
 'class' => 'ajax'  
 ]  
 );  
 **return parent**::*\_prepareLayout*();  
}

* Trong helper
* Trong file này chứa các phương thức tự định nghĩa

**<?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**]);  
 }  
}

### Create Products grid controllers

Tạo controller cho product tab

<?*php***namespace** Webspeaks\ProductsGrid\Controller\Adminhtml\Contacts;  
  
**use** Magento\Backend\App\Action;  
**use** Magento\TestFramework\ErrorLog\Logger;  
  
**class** Products **extends** \Magento\Backend\App\Action  
{  
  
 */\*\*  
 \** ***@var*** *\Magento\Framework\View\Result\LayoutFactory  
 \*/* **protected** $\_resultLayoutFactory;  
  
 */\*\*  
 \** ***@param*** *\Magento\Framework\View\Result\LayoutFactory $resultLayoutFactory  
 \** ***@param*** *Action\Context $context  
 \*/* **public function** \_\_construct(  
 Action\Context $context,  
 \Magento\Framework\View\Result\LayoutFactory $resultLayoutFactory  
 ) {  
 **parent**::*\_\_construct*($context);  
 $this->\_resultLayoutFactory = $resultLayoutFactory;  
 }  
  
 */\*\*  
 \* {****@inheritdoc****}  
 \*/* **protected function** \_isAllowed()  
 {  
 **return true**;  
 }  
  
 */\*\*  
 \* Save action  
 \*  
 \** ***@return*** *\Magento\Framework\Controller\ResultInterface  
 \*/* **public function** execute()  
 {  
 $resultLayout = $this->\_resultLayoutFactory->create();  
 $resultLayout->getLayout()->getBlock('wsproductsgrid.edit.tab.products')  
 ->setInProducts($this->getRequest()->getPost('contact\_products', **null**));  
  
 **return** $resultLayout;  
 }  
}

Tạo controller cho grid form product

<?*php***namespace** Webspeaks\ProductsGrid\Controller\Adminhtml\Contacts;  
  
**use** Magento\Backend\App\Action;  
**use** Magento\TestFramework\ErrorLog\Logger;  
  
**class** ProductsGrid **extends** \Magento\Backend\App\Action  
{  
  
 */\*\*  
 \** ***@var*** *\Magento\Framework\View\Result\LayoutFactory  
 \*/* **protected** $\_resultLayoutFactory;  
  
 */\*\*  
 \** ***@param*** *\Magento\Framework\View\Result\LayoutFactory $resultLayoutFactory  
 \** ***@param*** *Action\Context $context  
 \*/* **public function** \_\_construct(  
 Action\Context $context,  
 \Magento\Framework\View\Result\LayoutFactory $resultLayoutFactory  
 ) {  
 **parent**::*\_\_construct*($context);  
 $this->\_resultLayoutFactory = $resultLayoutFactory;  
 }  
  
 */\*\*  
 \* {****@inheritdoc****}  
 \*/* **protected function** \_isAllowed()  
 {  
 **return true**;  
 }  
  
 */\*\*  
 \* Save action  
 \*  
 \** ***@return*** *\Magento\Framework\Controller\ResultInterface  
 \*/* **public function** execute()  
 {  
 $resultLayout = $this->\_resultLayoutFactory->create();  
 $resultLayout->getLayout()->getBlock('wsproductsgrid.edit.tab.products')  
 ->setInProducts($this->getRequest()->getPost('contact\_products', **null**));  
  
 **return** $resultLayout;  
 }  
  
}

Tạo layout cho controller

[/view/adminhtml/layout/wsproductsgrid\_contacts\_products.xml](https://github.com/arvind07/productsgrid_magento2/blob/master/Webspeaks/ProductsGrid/view/adminhtml/layout/wsproductsgrid_contacts_products.xml)

<?xml version="1.0" encoding="UTF-8"?>  
<layout xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="urn:magento:framework:View/Layout/etc/layout\_generic.xsd">  
<container name="root" label="Root">  
 <block class="Webspeaks\ProductsGrid\Block\Adminhtml\Contact\Edit\Tab\Products" name="wsproductsgrid.edit.tab.products"/>  
 <block class="Magento\Backend\Block\Widget\Grid\Serializer" name="products\_grid\_serializer">  
 <arguments>  
 <argument name="grid\_block" xsi:type="string">wsproductsgrid.edit.tab.products</argument>  
 <argument name="callback" xsi:type="string">getSelectedProducts</argument>  
 <argument name="input\_element\_name" xsi:type="string">products</argument>  
 <argument name="reload\_param\_name" xsi:type="string">contact\_products</argument>  
 </arguments>  
 </block>  
</container>  
</layout>

[/view/adminhtml/layout/wsproductsgrid\_contacts\_productsgrid.xml](https://github.com/arvind07/productsgrid_magento2/blob/master/Webspeaks/ProductsGrid/view/adminhtml/layout/wsproductsgrid_contacts_productsgrid.xml)

<?xml version="1.0" encoding="UTF-8"?>  
<layout xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="urn:magento:framework:View/Layout/etc/layout\_generic.xsd">  
<container name="root" label="Root">  
 <block class="Webspeaks\ProductsGrid\Block\Adminhtml\Contact\Edit\Tab\Products" name="wsproductsgrid.edit.tab.products"/>  
</container>  
</layout>

Tạo block cho noi dung hiên thị tab

<?*php***namespace** Webspeaks\ProductsGrid\Block\Adminhtml\Contact\Edit\Tab;  
  
**use** Webspeaks\ProductsGrid\Model\ContactFactory;  
  
**class** Products **extends** \Magento\Backend\Block\Widget\Grid\Extended  
{  
 */\*\*  
 \** ***@var*** *\Magento\Catalog\Model\ResourceModel\Product\CollectionFactory  
 \*/* **protected** $productCollectionFactory;  
  
 */\*\*  
 \* Contact factory  
 \*  
 \** ***@var*** *ContactFactory  
 \*/* **protected** $contactFactory;  
  
 */\*\*  
 \** ***@var*** *\Magento\Framework\Registry  
 \*/* **protected** $registry;  
  
 **protected** $\_objectManager = **null**;  
  
 */\*\*  
 \*  
 \** ***@param*** *\Magento\Backend\Block\Template\Context $context  
 \** ***@param*** *\Magento\Backend\Helper\Data $backendHelper  
 \** ***@param*** *\Magento\Framework\Registry $registry  
 \** ***@param*** *ContactFactory $attachmentFactory  
 \** ***@param*** *\Magento\Catalog\Model\ResourceModel\Product\CollectionFactory $productCollectionFactory  
 \** ***@param*** *array $data  
 \*/* **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);  
 }  
  
 */\*\*  
 \* \_construct  
 \** ***@return*** *void  
 \*/* **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('contact\_id')) {  
 $this->setDefaultFilter(**array**('in\_product' => 1));  
 }  
 }  
  
 */\*\*  
 \* add Column Filter To Collection  
 \*/* **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;  
 }  
  
 */\*\*  
 \* prepare collection  
 \*/* **protected function** \_prepareCollection()  
 {  
 $collection = $this->productCollectionFactory->create();  
 $collection->addAttributeToSelect('name');  
 $collection->addAttributeToSelect('sku');  
 $collection->addAttributeToSelect('price');  
 $this->setCollection($collection);  
 **return parent**::*\_prepareCollection*();  
 }  
  
 */\*\*  
 \** ***@return*** *$this  
 \*/* **protected function** \_prepareColumns()  
 {  
 */\** ***@var*** *$model \Webspeaks\ProductsGrid\Model\Slide \*/* $model = $this->\_objectManager->get('\Webspeaks\ProductsGrid\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*();  
 }  
  
 */\*\*  
 \** ***@return*** *string  
 \*/* **public function** getGridUrl()  
 {  
 **return** $this->getUrl('\*/\*/productsgrid', ['\_current' => **true**]);  
 }  
  
 */\*\*  
 \** ***@param*** *object $row  
 \** ***@return*** *string  
 \*/* **public function** getRowUrl($row)  
 {  
 **return** '';  
 }  
  
 **protected function** \_getSelectedProducts()  
 {  
 $contact = $this->getContact();  
 **return** $contact->getProducts($contact);  
 }  
  
 */\*\*  
 \* Retrieve selected products  
 \*  
 \** ***@return*** *array  
 \*/* **public function** getSelectedProducts()  
 {  
 $contact = $this->getContact();  
 $selected = $contact->getProducts($contact);  
  
 **if** (!is\_array($selected)) {  
 $selected = [];  
 }  
 **return** $selected;  
 }  
  
 **protected function** getContact()  
 {  
 $contactId = $this->getRequest()->getParam('contact\_id');  
 $contact = $this->contactFactory->create();  
 **if** ($contactId) {  
 $contact->load($contactId);  
 }  
 **return** $contact;  
 }  
  
 */\*\*  
 \* {****@inheritdoc****}  
 \*/* **public function** canShowTab()  
 {  
 **return true**;  
 }  
  
 */\*\*  
 \* {****@inheritdoc****}  
 \*/* **public function** isHidden()  
 {  
 **return true**;  
 }  
}