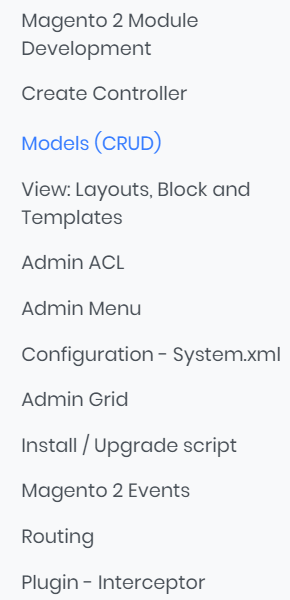
# 03 - Module sample

[link source code](https://gitlab.com/pirago/magento-demo) \*\*Đọc các bài hướng dẫn cơ bản [tại đây](https://www.mageplaza.com/magento-2-module-development/) 

## Yêu cầu

clone lại module Sizechart từ Mageplaza, tổng quan, giới thiệu và demo xem [tại đây](https://www.mageplaza.com/magento-2-size-chart/)

## Tổng quát các bước

1. Đăng kí, khởi tạo module
2. Tạo CSDL, Model, Resource Model và Collection để thao tác với dữ liệu
3. Menu & Configuration
4. Quản lý Rule trong admin
5. Tạo nhiều storeview để test
6. Logic hiển thị sizechart cho 3 kiểu hiển thị (popup, inline, tab)

## Đăng kí, khởi tạo module

Tạo một module mới với tên PiraGo\_SizeChart, version 1.0.0. \* PiraGo là *namespace / company / vendor name* \* SizeChart là tên module

Thư mục: *app/code/PiraGo/SizeChart*

1. tạo file *app/code/PiraGo/SizeChart/registration.php:*
2. tạo file *app/code/PiraGo/SizeChart/etc/module.xml:*

\*\*node sequence thể hiện dependency của module core Magento\_Catalog

## Tạo CSDL, Model, Resource Model và Collection để thao tác với dữ liệu

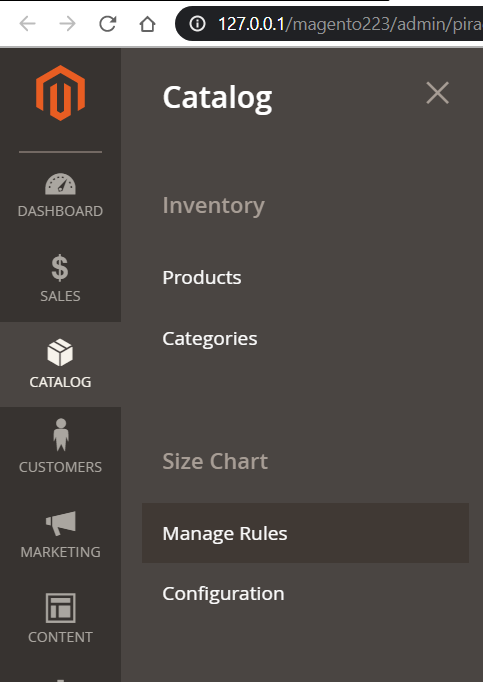
1. **Tạo bảng CSDL**

Tạo bảng tên "sizechart\_rules" trong phpmyadmin để lưu trữ danh sách rule. Sau đó tạo file setup bảng (*app/code/PiraGo/SizeChart/Setup/InstallSchema.php*) Bảng gồm các cột sau:  **file InstallSchema chỉ chạy một lần lúc setup module nên muốn thêm cột hoặc data thì phải tạo file upgradeSchema, lưu ý đến thống số setup\_version trong file module.xml khi upgrade tìm hiểu** [**ở đây**](https://www.mageplaza.com/magento-2-module-development/magento-2-how-to-create-sql-setup-script.html) **2.** Tạo CRUD Model (Create, Read, Update, Delete)kiến thức về ORM tìm hiểu [ở đây](https://blog.magestore.com/magento-orm/) Tạo model, resource model, collection để thao tác với dữ liệu trong bảng sizechart\_rules: \* *PiraGo.php* \* *PiraGo.php* \* *PiraGo.php*

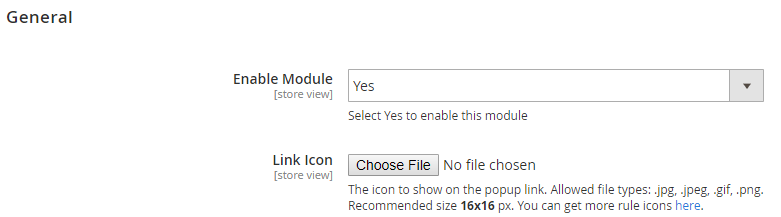
**Resource Model trực tiếp thực hiện truy vấn đến bảng và định nghĩa Model nào sẽ kết nối với bảng đó để thao tác, Collection Model coi như là Resurce Model cho phép truy vấn và lọc nhiều hàng trong bảng. Tìm hiểu thêm** [**ở đây**](https://www.mageplaza.com/magento-2-module-development/how-to-create-crud-model-magento-2.html) **3.** Tạo Route controller để test Model trênkiến thức về Routing tìm hiểu [ở đây](https://www.mageplaza.com/magento-2-module-development/magento-2-routing.html) đường dẫn controller theo cú pháp sau front\_name/controller\_name/action\_name Để test truy vấn qua Model vào phpmyadmin tạo một vài dòng dữ liệu. Sau đó tạo Controller: *PiraGo.php*

## Menu & Configuration

1. **Admin menu**

 \* Menu parent: Size Chart dưới đó có 2 menu con

+Manage Rules: Dẫn tới trang quản lý Rule (Grid). Action: sizechartạo file *PiraGo/SizeChart/etc/adminhtml/menu.xml* +Configuration: Dẫn tới trang cấu hình của module SizeChart (Configuration sẽ làm ở phần sau). Action adminhtml/system\_config/edit/section/sizechart \* Tạo resource acl (access control list)

+tạo file *PiraGo/SizeChart/etc/acl.xml* **là bước tạo resource cho cây acl như là tạo các quyền để phân quyền truy cập cho các user khác nhau, attribute 'resource' của node trong menu.xml là để gán quyền vào menu. Tức user có quyền đó mới truy cập được menu đó. 2.** Configuration\*\* 

Tạo trang cấu hình cho module (system.xml) với các field sau: \* Enable SizeChart (yes/no): bật/tắt tính năng sizechart. Nếu field này là NO, toàn bộ các tính năng của module ngoài frontend sẽ không hoạt động. \* Link icon: cấu hình image cạnh popup link ở frontend

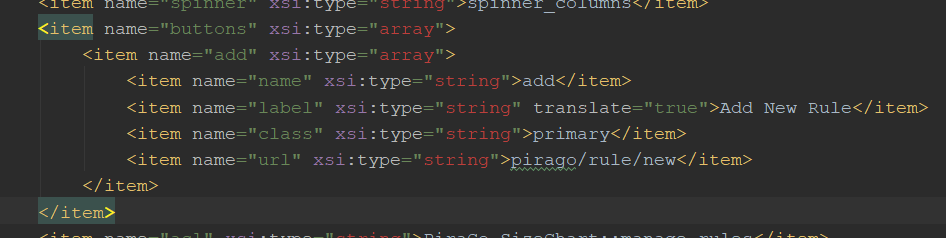
tạo file *PiraGo/SizeChart/etc/adminhtml/system.xml* \*\*node để cấu hình model thao tác với dữ liệu kiểu image, image sẽ được lưu trong thư mục được cấu hình trong node . Khi lấy đường dẫn ảnh để hiển thị thì dùng storeManager->getStore()->getBaseUrl. Xem function getPath() ở file PiraGo/SizeChart/Block/SizeChart1.php

## Quản lý Rule trong admin

1. \*\* Tạo Grid quản lý SizeChart Rules\*\*

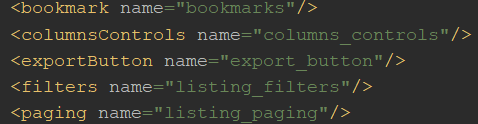
-Action: sizechart\rule``` +tạo layout file *PiraGo/SizeChart/view/adminhtml/layout/pirago*sizechart\_rule\_index.xml\_ +tạo ui component layout file *PiraGo/SizeChart/view/adminhtml/ui*component/pirago\_sizechart\_rule\_edit.xml\_

``` -Sử dụng UI Component tạo grid để quản lý SizeChart Rule với các function cơ bản: \* Nút New Rule trang form tạo rules. Action: sizechart\rule

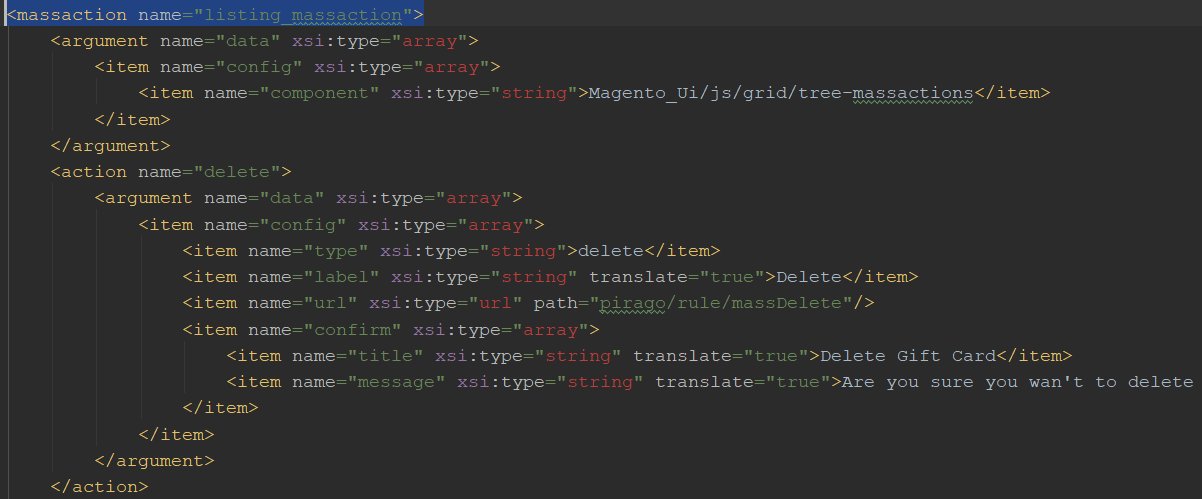


cấu hình nút add new rule

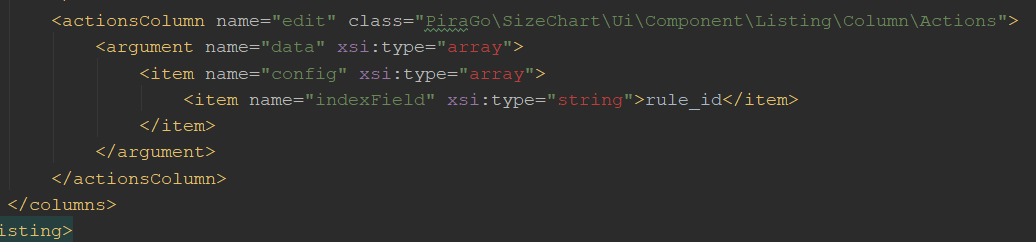
* Cấu hình các nút ở dưới nút Add New Rule

 để lọc nhanh dữ liệu để phân trang để bật tắt hiển thị các cột \* Các massActions: Delete, Change Status

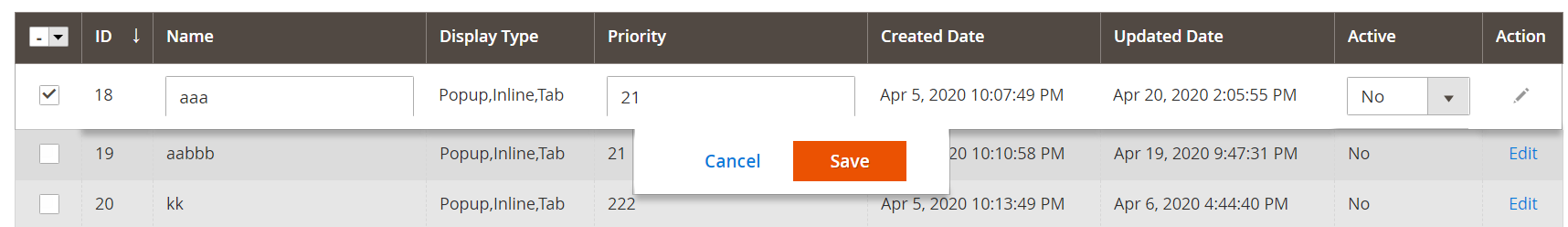
cấu hình trong node

 Các controller handle các massAction trên: pirago/rule/massDelete và pirago/rule/massStatus

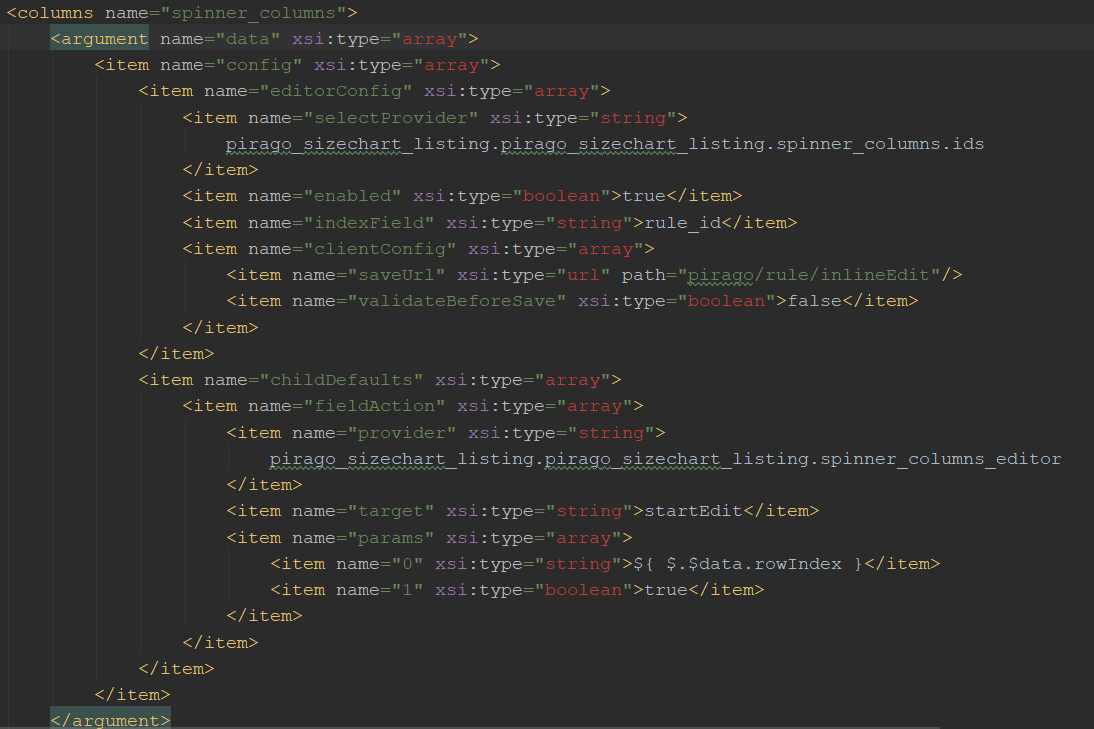
* Edit column: Mỗi row sẽ có action edit để dẫn tới trang edit rules đó - sizechart\rule*PiraGo.php* thêm một cột action vào grid

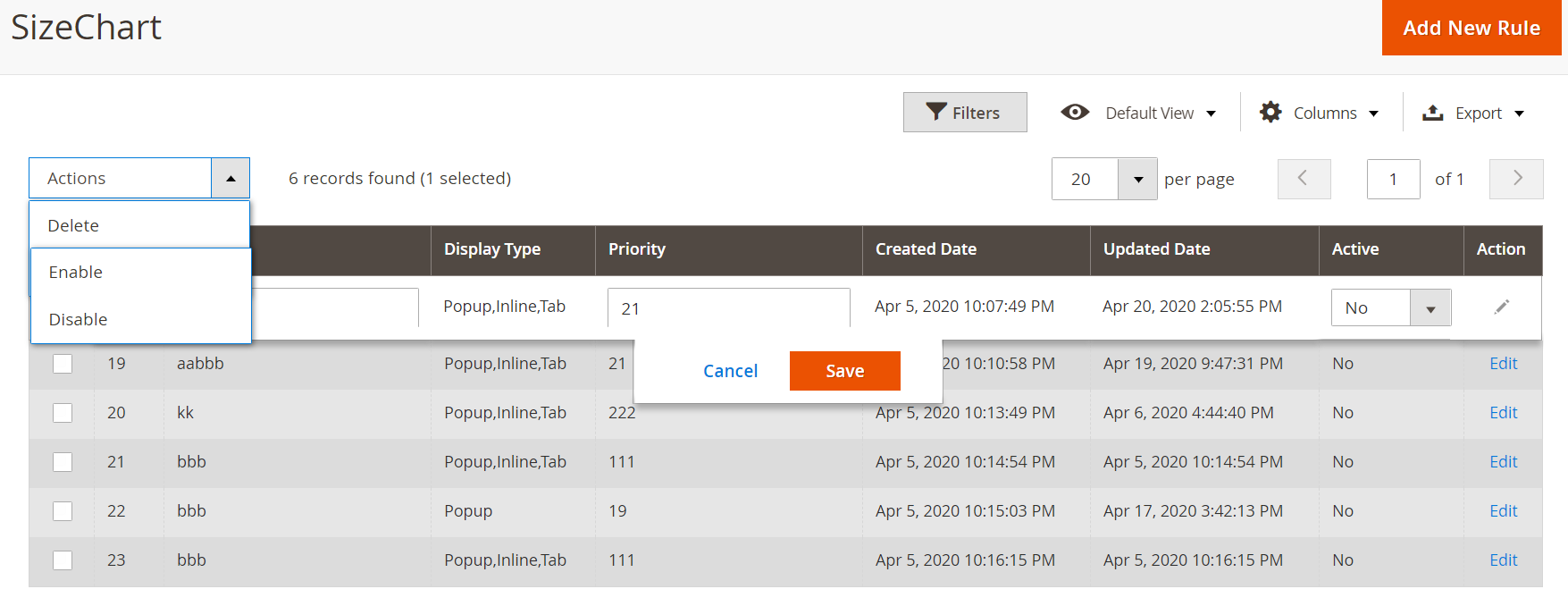


* Inline Edit: admin có thể sửa nhanh một cột trong grid



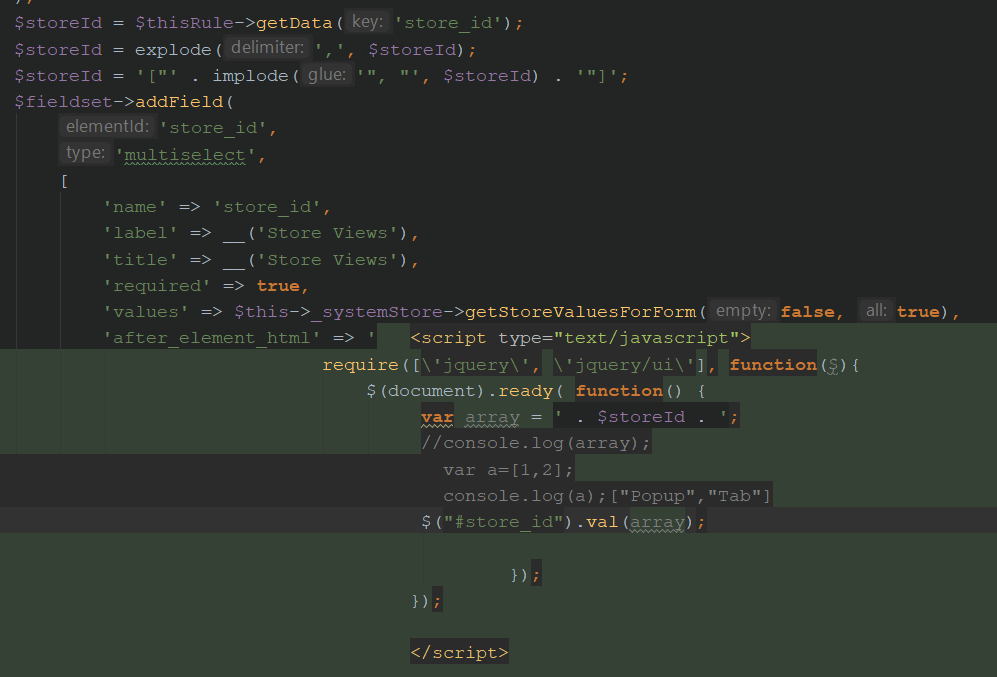
inline edit

 \*\*cấu hình trong node controller pirago/rule/inlineEdit handle action này \* xong cấu hình grid

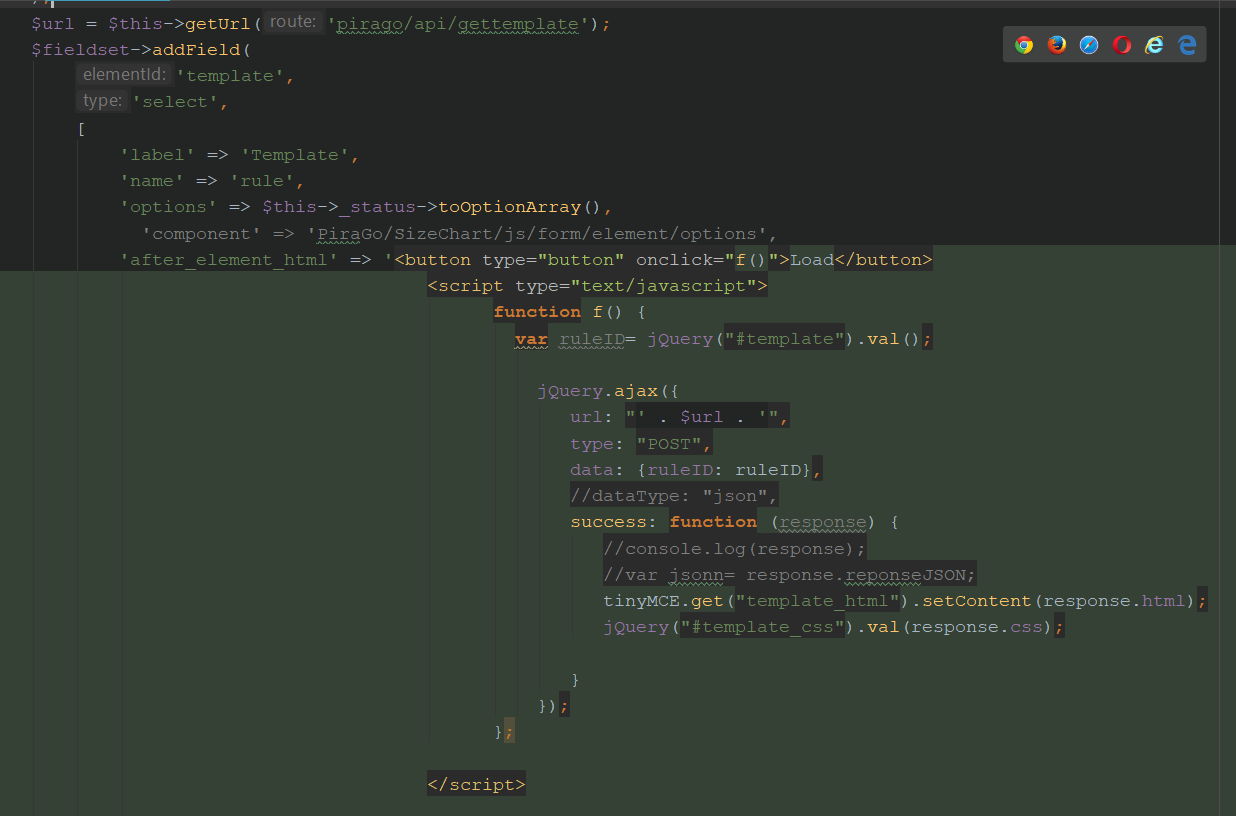


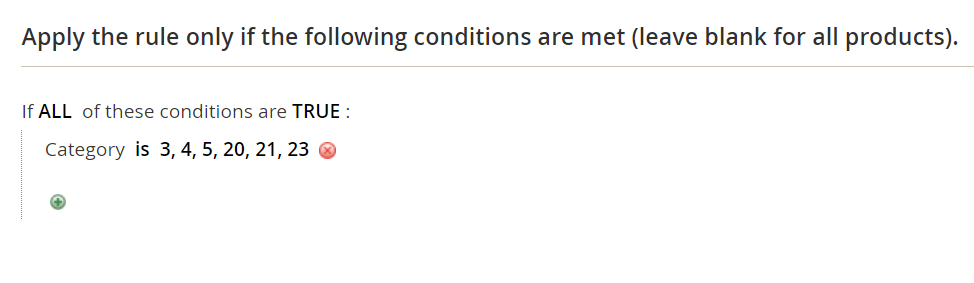
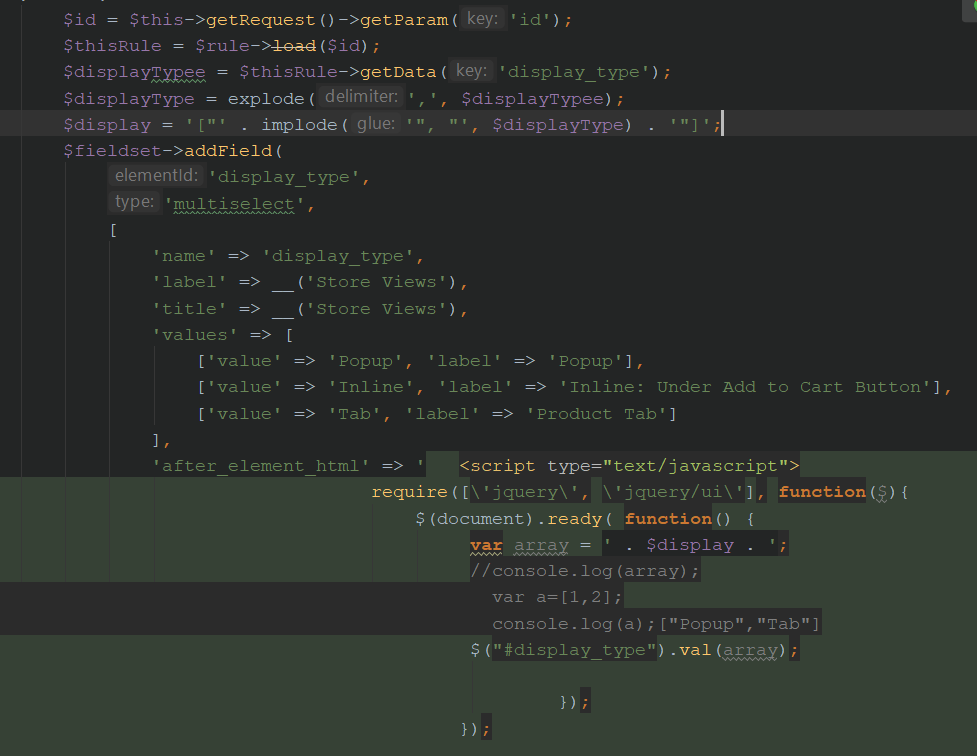
**tìm hiểu thêm** [**tại đây**](https://www.mageplaza.com/magento-2-module-development/create-admin-grid-magento-2.html) **2.** Tạo Form & Save Rules **\*** Tạo Form\*\*

\*\*có 2 trường hợp tạo new rule hoặc edit rule. New rule thì các field sẽ trắng, còn edit rule thì các field được đổ vào dữ liệu tương ứng với rule id.  
+đầu tiên tạo layout handle action sizechart/rule/edit. Vì cấu hình form gồm 4 tab nên trong đó khai báo các block tương ứng vơí mỗi tab khác nhau. Mỗi tab gồm một vài field. Chú ý đến các node thể hiện ghi đè lên layout core của magento +tiếp theo tạo Block/Adminhtml/Code/Edit.php: thể hiện khai báo block của form container: trong file có câu lệnh thêm nút Save and Continute Edit. +tiếp theo tạo Block/Adminhtml/Code/Edit/Tab.php: khai báo block tab lớn bên trái layout editing page. +tiếp tạo Block/Adminhtml/Code/Edit/Form.php: khai báo thông tin form như id, action form. +Block/Adminhtml/Code/Edit/Tab/Code.php: tạo field trong tab con thứ 1 qua hàm addFieldset(). Chú ý đến các kiểu field. Trong tab này có field Storeview là multiselect thì key 'values' sẽ chỉ đến giá trị trong core để lấy ra các option về storeview. Nếu là form edit, đổ dữ liệu field dạng multiselect kiểu array phải chuyển php array sang js array



đổ dữ liệu dạng mutiselect

+Block/Adminhtml/Code/Edit/Tab/What.php: block này có field Template html là field dạng WYSIWYG editor cho phép soạn siêu văn bản. Ở tab này có thêm một button Load để call ajax load dữ liệu đổ vào field Template html. Thêm đoạn code cho nút Load trong key " after\_element\_html"  tinyMCE.get().setContent() để convert script html thành giao diện. +Block/Adminhtml/Code/Edit/Tab/Conditions.php: có field tạo dữ liệu kiểu condition cho rule. Để đọc và ghi được field này class model phải khai bào các dependency qua contruct thêm các class , \*để render thêm các conditions khi ấn vào dấu cộng phải có action sizechart/rule/NewConditionHtml.php

 +Block/Adminhtml/Code/Edit/Tab/How.php: tab này có field kiểu mutiselect phải convert sang string mới lưu được vào database. Khi edit form để render ra mutilselect field thì phải chuyển string -> php array ->js array.  \* **Save Rules**

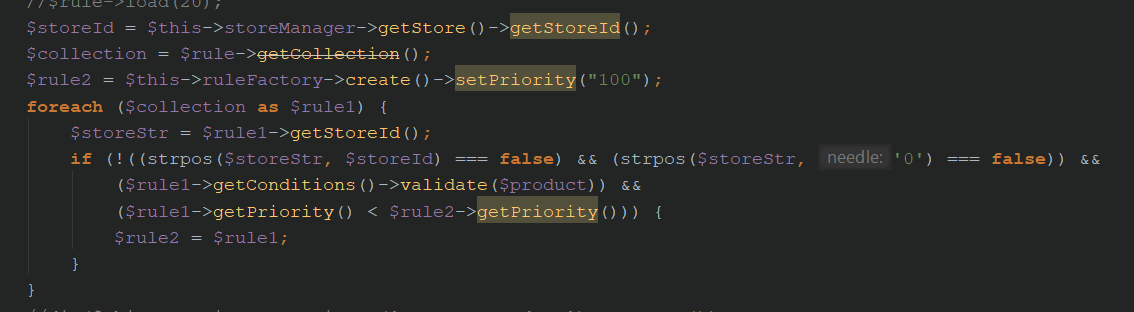
Action: sizechart/rule/Save Logic save rules: Để lưu conditions dùng method LoadPost() nên các field kiểu multiselect phải lưu trước vì phải qua bước convert sang String mới lưu được. Hàm prepareData() thao tác để lưu conditions. Logic chuyển hướng sau khi lưu: nếu ấn nút Save and Continue Edit thì chuyển hướng lại trang edit với param id, còn ấn nút Save thì chuyển về trang grid.

## Tạo nhiều storeview để test

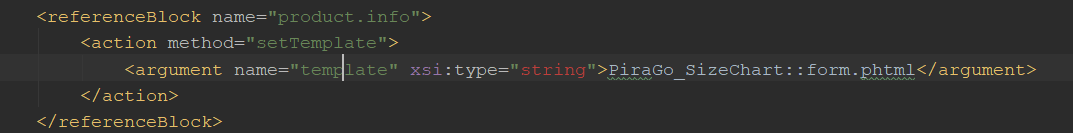
\*\*storeview khác nhau sẽ áp dụng rule khác nhau, nên tạo nhiều storeview để test <https://devdocs.magento.com/guides/v2.3/config-guide/multi-site/ms_websites.html>

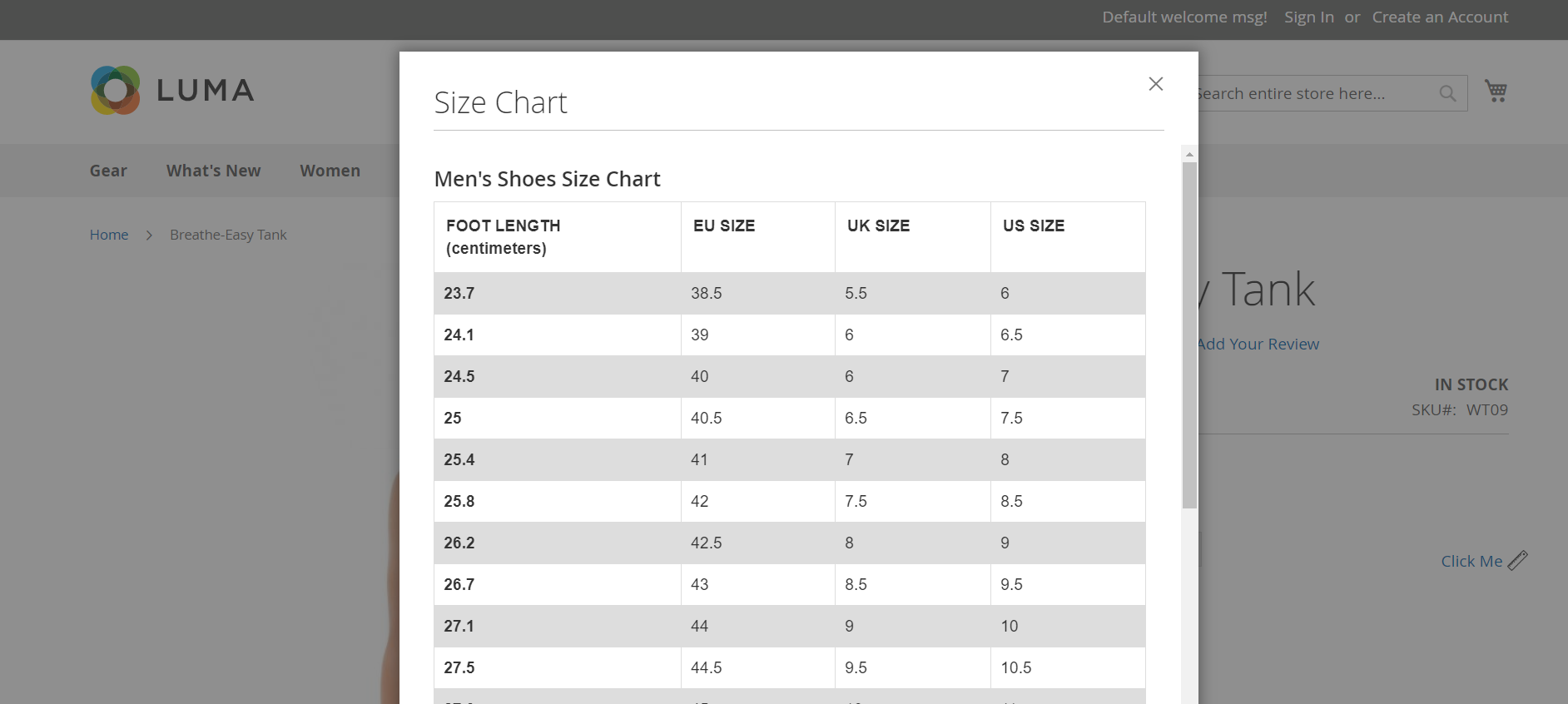
## Hiển thị sizechart cho 3 kiểu hiển thị (popup, inline, tab)

1. **Logic hiển thị**

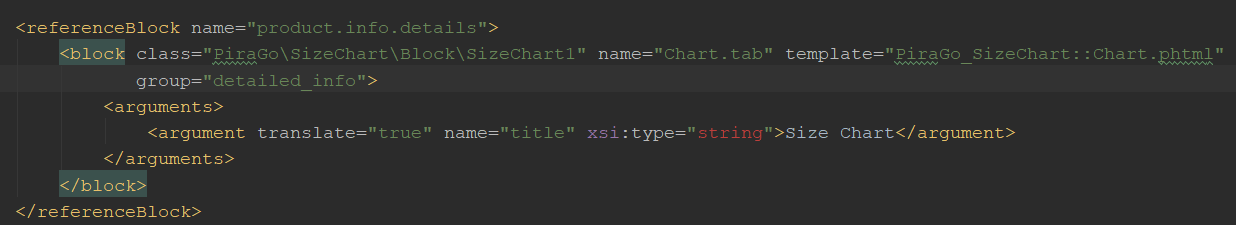
\*\* trước khi chạy vào block 'product view' thì tìm một rule thích hợp nhất được áp dụng, **hình dưới thể hiện logic đó: lặp tìm kiếm tất cả các rule nếu rule và product thỏa mãn về storeview, điều kiện conditions thì xét đến thuộc tính priority để lấy ra rule ưu tiên nhất.** file PiraGo/SizeChart/Observer/DisplayType.php : là class observer xử lí event layout\_generate\_blocks\_before sẽ chạy ngay trước khi block được chạy  **sau khi chọn được rule thích hợp nhất thì sẽ truyền đi những dữ liệu cần thiết để hiện thị rule đó ra màn hình như kiểu hiện thị là gì? template html? template css?** các giá trị sẽ truyền thông qua Registry để đưa đến các block tương ứng xử lí là các block SizeChart1 và Inline

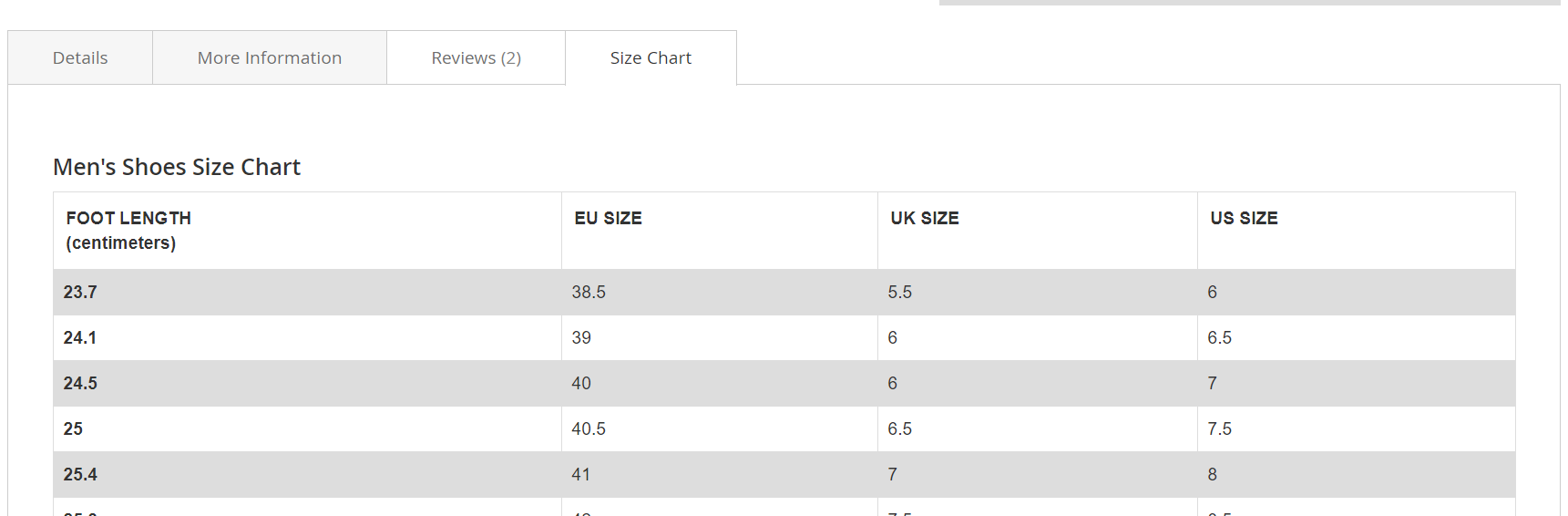
 2. **Override block layout cho các kiểu hiển thị** file: PiraGo/SizeChart/view/frontend/layout/catalog\_product\_view.xml \*\*để hiển thị ra frontend phải tìm đến những layout core để ghi đè lên, trong trường hợp này là layout catalog\_product\_view \* hiển thị kiểu popup: hình dưới thể hiện thêm action setTemplate() có nhiệm vụ thêm template "PiraGo\_SizeChart::form.phtml "vào block "product.info"





* hiển thị kiểu tab: block SizeChart1 extend block core là để thêm vài function về logic hiển thị

 node trong di.xml thể hiện block SizeChart1 override block Product

 \* hiển thị kiểu inline: tương tự node thể hiện block Inline được thêm vào Container đó kèm theo là template Inline.phtml chứa nội dung hiển thị

