Tạo SYSTEM.XML

* Dùng để tạo một số tab cần thiết trong Store->configuartion

**+** tạo system.xml trong adminhtml

--- system.xml---

+ tab :

. id : id cua tab

. translate : ten muc can dich thuong la label

. sortOrder : vi tri cua tab

+ section : nam trong tab, co the co nhieu group hoac nhieu con

. id

. transtate

. sortOrder

. showDefault

+ class : gia tri nay se ddc add nhu 1 class <class>separator-top123</class> : ten class la separator-top123

+ label :

+ tab : no la id cua tab , no sẽ cho magento biết section này thuộc về tab nào , và sectiob này sẽ được lưu ở tab đó

+ resource : khai báo rule ACL thứ mà admin và user phải có để truy cập vào cấu hình này

+ group : là nơi chưa 1 vài fields hoặc thuộc tính giống như section

+ fields : là phần chính của trang này , nó sẽ lưu dữ liệu chúng ta muốn để thiết lập , và trong các field này , chúng ta sẽ chỉ tập tring vào thuộc tính type. Nó sẽ định nghĩa fields show ra cái gì , text, select , button hay cái j khác

* Sau đó tạo file config.xml trong etc với mục đích set dữ liệu cho section ta đã tạo trong system.xml
  + Default:
  + <id section>
  + <general>
  + ..<enable>1</enable> : chọn yes no
  + <display\_text> Hello</display\_text> : set text cho type text in system.xml

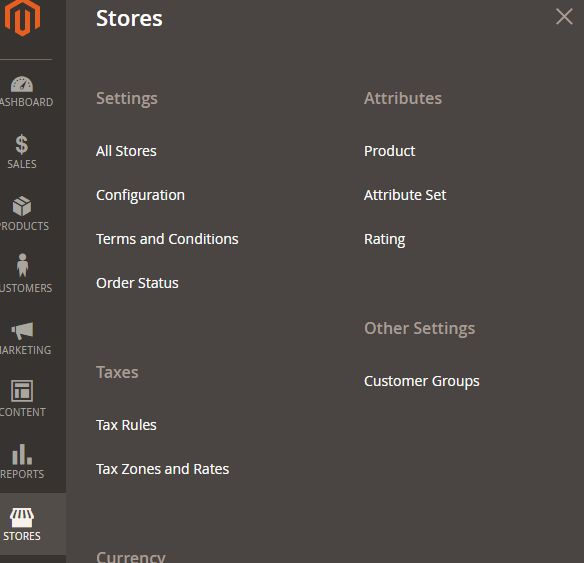
--------------------------------------------------------------------------------------------------------

Làm thế nào để tạo menu trong admin

: Mục đích : show 1 mục mới trong phần quản lí admin

1. Cấu trúc của menu admin là ntn?

Cấu trúc theo level, chúng ta sẽ thấy level 0 ở side bên trái



Chnugs ta có thể thấy STORE là level 0

Và tiếp theo là những level cao hơn

[*http://localhost/magento-simpledata/admin\_epqy0i/catalog/product\_attribute/index/key/9ffa5b5ba1ff6e746a443e896aeb0633d6a8504b792c01dec31147e06ebccc5c/*](http://localhost/magento-simpledata/admin_epqy0i/catalog/product_attribute/index/key/9ffa5b5ba1ff6e746a443e896aeb0633d6a8504b792c01dec31147e06ebccc5c/)

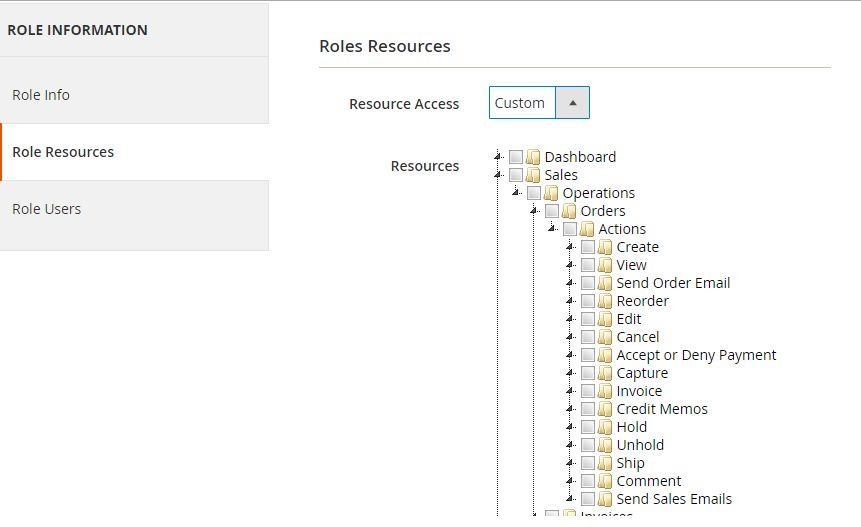
{router\_name}\_{controller\_folder}\_{action\_name}.

Đâu tiên ta sẽ tạo 1 menu level 0 có tên là “HelloWorld” và 2 sub-menu tên alaf “manage item” và “configuration” và menu.xml sẽ định nghĩa 1 collection of “add” : nó sẽ add 1 menu vào magento backend

* Id: vendor\_modulename :: menu\_description
* Title: là tên show ở menu
* Module : cho biêt menu thuộc về module nào
* Parent : có thể là id của menu khác, nó cho biết là menu này là con của menu khác, ví dụ parent=”Intersquad\_HelloWorld::hello”, nó hiểu manage item la còn của hello world menu , và sẽ show trong helloworld
* Action : khái báo url mà link trỏ tới với định dạng
  + {router\_name}*{controller\_folder}*{action\_name}.
* Resource : dùng để định danh access control lít (ACL)

**Tìm hiểu về Admin access control list**

+ vấn đề đặt ra là chủ sở hữu của của trang web Admin muốn tạo ra nhiều quyền truy cập khác nhau tùy theo nhân viên, ví dụ nhân viên A đc sửa sản phẩm nhưng không ddc thêm sp, nhân viên B đc thêm sp nhưng không đc xóa sp



Vào system -> user roles sẽ thấy như hình chúng ta có thể thấy một list cái các tài nguyên của hệ thống.

**----- TOPIC 9: CÁCH TẠO CÀI ĐẶT SQL TRONG MAGENTO 2**

**+** khi chúng ta muốn thay đổi cấu trúc database hay thêm 1 vài dữ liệu mới cho table, magento cung cấp 1 số lớp để thực hiện

* InstallSchema: sẽ chạy khi module được chạy để cài đặt cấu trúc database
* InstallData : sẽ chạy khi module đc chạy để khởi tạo dữ liệu cho database
* UpgradeSchema : nâng cấp để cài đặt cấu trúc database
* UpgradeData: add/remove dữ liệu cho table

Tất cả các lớp trên được chứa trong thư mục Module/Setup

Install và upgrade sẽ được chạy sao khi chạy lệnh setup:upgrade

InstallSchema và InstallData sẽ được chạy khi intall module

* InstallSchema:
  + Lớp này được extends từ InstallSchemaInterface
  + Lớp này phải có install() với 2 đối số SchemaSetupInterface và ModuleContextInterface​
    - SchemaSetupInterface: là một đối tượng cài đặt cung cấp 1 số hàm để tương tác với database
    - ​ModuleContextInterface: chỉ có 1 method là *getVersion*() sẽ return lại version hiện tại của module

InstallData sẽ chạy sau thằng InstallSchema

* UpgradeSchema / UpgradeData

2 file này sẽ được chạy khi install hoặc upgrade table

2 lớp này thì khác nhau ở lớp install bởi vì nó sẽ chạy mỗi khi upgrade module ,

Do đó ta cần phải check version

**Recuring**

Recuring là 1 script sẽ dc chạy sau khi cài đặt module sau mỗi lệnh setup:upgrade

Nó được khai báo giống như InstallSchema và chỉ khác ở cái tên /module-indexer/Setup/Recurring.php

**Uninstall**

Sẽ remove table, dữ liệu giống như nó chưa từng chạy…

* Cần cho nó biết table nào dropTable

**Tạo model**

+ model có các function khác khác nhau để quản lí data,

install cũng như là upgrade module,

ở đây chúng ta sẽ tạo Model . ResourceModel , ResourceModel Collection,

+ Model sẽ extends thằng abstractModel và implements thằng identityInterface( define thằng getIndentities() trả về 1 id duy nhất cho model)

Chúng ta chỉ dùng identityInterface nếu model yêu cầu refresh cache

sau khi tương tác database và hiển thị thông tin ra front end

\_\_construct() : sẽ được chạy bất cứ khi nào model khởi tạo.

Contruct sẽ gọi \_\_init define thằng resourceModel lấy thông tin từ database

const CACHE\_TAG = 'mageplaza\_helloworld\_post';

-> định danh duy nhất trong caching

protected $\_cacheTag = 'mageplaza\_helloworld\_post';

protected $\_eventPrefix = 'mageplaza\_helloworld\_post';

-> tiền tố cho sự kiện đc kích hoạt

**ResourceModel**

**+** chứa truy vấn sql,

+ extends ABSTRACTBD chứa các hàm lấy thông tin từ database

+ \_contruct sẽ gọi \_init chứa database và key

**ResourceModelCollection**

**+ cho phép chúng ta lọc cũng như lấy thông tin từ database**

**+** AbstractCollection : khởi tạo model , resource model trong construct

Factory Object

Sau khi tạo model , ta sẽ nói về FACTORY OBJECT , trong OOP một factory dùng để khỏi tạo 1 object

+ Name của Factory là name của Lớp Model + Factory

+ bất cứ lớp nào có tên Factory nó sẽ tự dộng sinh ra 1 lớp Factory trong var/generation

**-------------------ROUTING---------------------**

**+** Route sẽ định danh name cho 1 module chúng ta có thể dùng trên url để tìm module và execute phương thức action

**+ front controller : chay vong lap de tim ra route hop li vs rquét neu tim dc route dung thi break**

* **Grid trong admin la gi?**
  + **La 1 loai cua table liet ke ra cac items trong database cung cap 1 so thuoc tinh sort , filter , delete, update**

-----------INDEXER-------------

+ tạo custom indexer

* Các kiểu indexing
* \* full indexing
  + Full-reindexing build lại toàn bộ bảng database liên quan đến indexing
  + Full indexing có thể được gây ra bởi nhiều lí do bao gồm cả việc tạo web store mới ,hoặc nhóm khách hàng mới
  + Cài đặt full indexing bằng conmmand line
* \* partial indexing
  + Build lại bảng trong database cho việc thay đổi ví dụ như thay đổi attribute hay giá của sản phẩm
* Index modes : 2 mode
  + Update on save : đc updata ngay lập tức ngay sau khi dữ liệu thay đổi
  + Updata on schedule: update bởi cron job tùy vào cấu hình
* Index status
  + Valid
  + Invalid
  + Working

**CÁCH TẠO 1 INDEX**

* Indexer.xml : define indexer
  + Id : được dùng để định danh định danh indexer, có thể dùng để gọi nó khi bạn muốn check status, mode , reindex
  + View\_id : id của mview
  + Class : tên của class mà chúng ta sử lí indexer
  + Title
  + description
* Mview.xml: theo dõi thay đổi của database cho 1 entity nhất định và chạy execute
  + Id: được gọi từ indexer và class chứa execute: nó sẽ chạy khi table đăng kí bị thay đổi
* Indexer class
  + Execute() sẽ đc thực thi data thay đổi

------------ EVENTS AND OBSERVERS -------

+ events và observers thực hiện magento 2 dựa trên public subcribe pattern , có thể sử dụng code của mình để trả về 1 sự kiện của magento cụ thể hoặc thậm chí sự kiện mình về

Events: được gửi đi bởi module khi 1 hành động cụ thể đc kich hoạt, ngoài ra có thể là sự kiên của chính nó, magento cho phép bạn tự tạo events đc gửi đi trong code của bạn,khi 1 events đc gửi đi, bạn có thế pass data to any observer để xem event

* Gửi event
  + T mún gửi 1 danh sách các event cho phép các module khác có thể thay đổi từ helloWorld, t sẽ thay dổi controller như sau

Ta sẽ nhận 2 tham số argument là tên event và 1 mãng dữ liệu,ta thêm data object vào event và gọi nó trở lại

Events.xml

Chúng ta định danh 1 event với tên là tên event được gửi ở trên

và observer sẽ thực thi ,