# Phần 2

## Bài 1 cài đặt moi trường lập trình

File đính kèm

* [config.yaml](https://video1.techmaster.vn/stream/attach_files/download/1400/YJAfJwoe65ALIiS2ql6tujYllEDyl3HaIy6hKxlUE7jmw4nIK5/eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJwcmljZSI6MCwiZnVsbCI6dHJ1ZSwidmlkZW8iOnsiaWQiOjEwODAsIm5hbWUiOiJtYWdlbnRvLWNhaS1kYXQtbW9pLXRydW9uZy1sYXAtdHJpbmgubXA0IiwicGF0aCI6Ii9tZWRpYS8yNTUyMCJ9LCJhdHRhY2htZW50IjpbeyJpZCI6MTQwMCwibmFtZSI6ImNvbmZpZy55YW1sIiwicGF0aCI6Ii9tZWRpYS8yNTUyMCJ9LHsiaWQiOjE1MjgsIm5hbWUiOiJWYWdyYW50ZmlsZSIsInBhdGgiOiIvbWVkaWEvMjU1MjAifV19.H3SOyDsy2C5lq0k9LC6VnDkPAZtv56H0OGpl_IwZzP8)
* [Vagrantfile](https://video1.techmaster.vn/stream/attach_files/download/1528/YJAfJwoe65ALIiS2ql6tujYllEDyl3HaIy6hKxlUE7jmw4nIK5/eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJwcmljZSI6MCwiZnVsbCI6dHJ1ZSwidmlkZW8iOnsiaWQiOjEwODAsIm5hbWUiOiJtYWdlbnRvLWNhaS1kYXQtbW9pLXRydW9uZy1sYXAtdHJpbmgubXA0IiwicGF0aCI6Ii9tZWRpYS8yNTUyMCJ9LCJhdHRhY2htZW50IjpbeyJpZCI6MTQwMCwibmFtZSI6ImNvbmZpZy55YW1sIiwicGF0aCI6Ii9tZWRpYS8yNTUyMCJ9LHsiaWQiOjE1MjgsIm5hbWUiOiJWYWdyYW50ZmlsZSIsInBhdGgiOiIvbWVkaWEvMjU1MjAifV19.H3SOyDsy2C5lq0k9LC6VnDkPAZtv56H0OGpl_IwZzP8)

**Link download các phần mềm cần thiết**

- Git: <https://git-scm.com/downloads> (có kèm theo SSH Tool và Git Bash giúp gõ lệnh command line + ssh trên Windows)

- VirtualBox: <https://www.virtualbox.org/wiki/Downloads>

- Vagrant: <https://www.vagrantup.com/downloads.html>

- PHPStorm: <https://www.jetbrains.com/phpstorm/download/> (IDE dùng để lập trình PHP)

**Lệnh bật báo lỗi trong môi trường Development (ở Production cần bỏ đi)**

ini\_set('display\_errors', 1);

ini\_set('display\_startup\_errors', 1);

error\_reporting(E\_ALL);

**Chú ý**

- Nếu khi chạy **vagrant up** gặp lỗi *Box 'ubuntu/trusty64' could not be found* thì chạy lệnh sau:

vagrant box add ubuntu/trusty64 https://atlas.hashicorp.com/ubuntu/boxes/trusty64

- Để bật máy ảo ta dùng lệnh **vagrant up** còn tắt máy ảo thì dùng lệnh **vagrant halt**. Ngoài ra để kết nối vào máy ảo ta dùng lệnh **vagrant ssh.** Máy ảo chỉ đóng vai trò làm server, còn lập trình thì vẫn dùng IDE trên máy local. Có thể tạo các **snapshot** cho máy ảo bằng Virtualbox hoặc lệnh vagrant để có thể restore khi server bị lỗi (do cài cắm sai hoặc code sai), tham khảo [Vagrant Snapshot](https://www.vagrantup.com/docs/cli/snapshot.html)

- Trong trường hợp máy quá yếu không cài được máy ảo thì có thể cài magento trên XAMPP, tuy nhiên khi code sẽ có thể hơi khác 1 chút so với trong video vì môi trường lập trình khác. Do đó nếu gặp lỗi liên quan đến cài đặt các bạn sẽ phải tự tra cứu trên mạng.

## Bài 2 module hello

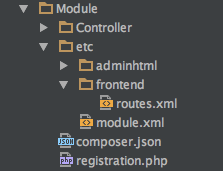
**Chú ý**

- Trong các ví dụ sử dụng phím tắt là **Ctrl** (trên Windows), trên Mac thì thay bằng phím **Command**.

- Một số thuật ngữ, ký hiệu dùng trong bài học (khóa học):

* **Core**: Mã nguồn chính của ứng dụng, chúng ta sẽ không chỉnh sửa trực tiếp để tránh bị lỗi
* **Vendor**: Nhà cung cấp, có thể là 1 công ty, tổ chức hay cá nhân nào đó. Ký hiệu Vendor ~ tên của Vendor, ví dụ trong bài là Robin
* **Module**: Là một chức năng có thể do Mangento viết ra (core) hoặc do 1 bên thứ 3 cung cấp (còn được gọi là **Extension**). Ký hiệu Module ~ tên của Module, ví dụ trong bài là Bai1
* **Router**: Là thành phần điều hướng, chuyển hướng người dùng truy cập từ 1 link nào đó vào code xử lý tương ứng

- Tóm tắt cấu trúc module đã học: (tham khảo cấu trúc module đầy đủ tại [Create your component file structure](http://devdocs.magento.com/guides/v2.0/extension-dev-guide/build/module-file-structure.html))



* **registration.php**: Đăng ký thư mục cài đặt cho module (dùng cho production). Mặc định module khi được cài đặt bằng composer sẽ ở trong thư mục <Thư mục cài magento>**/vendor/**
* **composer.json**: Định nghĩa các thành phần phụ thuộc và 1 số thông tin của module như: Tên, mô tả, version, ...
* **Controller**: Chứa các controller/action của module
* **etc**: Chứa các file cấu hình, trong đó có file module.xml là bắt buộc (cấu hình tên module, version cài đặt, version database, ...). Bên trong là các file cấu hình cho từng khu vực (adminhtml, frontend, ...) như: cấu hình routes, menu, ...

- Khi tạo module mới cho Magento 2 thì nên bắt đầu từ version 2.0.0 để đánh dấu là module cho Magento 2. Tham khảo thêm về cách đánh version tại [Versioning policy overview](http://devdocs.magento.com/guides/v2.1/architecture/versioning.html)

- Khi đặt tên router thì phải có ít nhất 3 ký tự và sử dụng các ký tự **a-z**, **A-Z**, **0-9**, **-**, **\_**

- Controller cho frontend thì kế thừa class**\Magento\Framework\App\Action\Action**(nằm trong thư mục **lib > internal > Magento > Framework > App > Action > Action.php**)

- Nếu đặt tên Controller hoặc Action là **Index** thì link truy cập có thể bỏ qua phần index. Ví dụ: Module Bai1, Controller Index, Action Index có thể truy cập bằng link /bai1/index/index hoặc chỉ cần /bai1 cũng được.

**Luyện tập**

- Tạo controller cho các route sau:

* [router\_id]**/**
* [router\_id]**/helloworld**
* [router\_id]**/hello/world**

*Source code ví dụ mẫu* <https://github.com/robinhuy/magento2-example/tree/v1.0>

## Bài 3

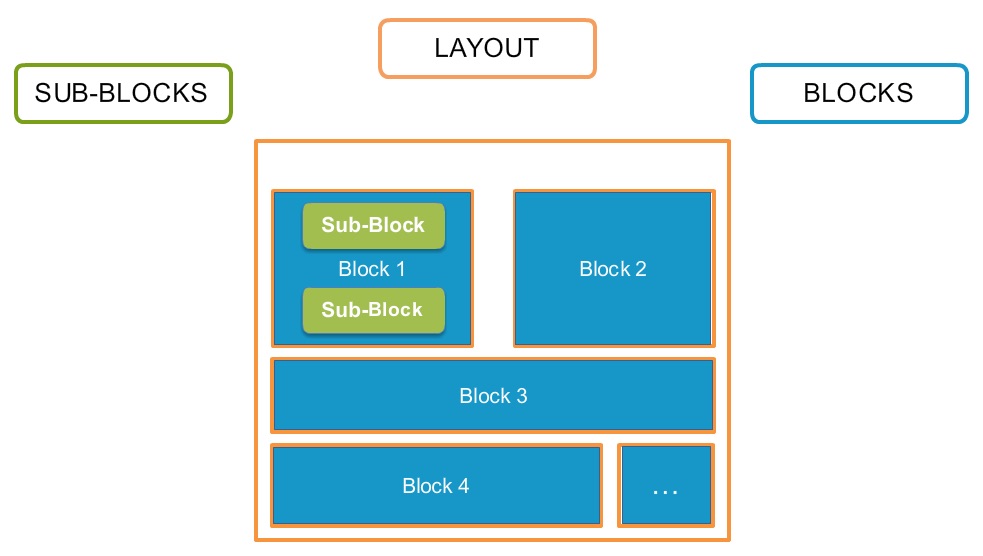
Tham khảo cấu hình layout bằng XML: <http://devdocs.magento.com/guides/v2.1/frontend-dev-guide/layouts/xml-manage.html>

Code mẫu khai báo block:

<**referenceBlock** name="content">

<**block** class="Robin\Bai1\Block\Weather" name="robin\_bai1" template="weather.phtml" />

</**referenceBlock**>

Mô hình mô tả Layout và Block

**Chú ý**

- Block có thể được dùng ở các Module khác nhau, và có thể có các Template khác nhau để tái sử dụng code.

- Function **execute()** của Controller sẽ phải trả về 1 trong các loại **"result object"**sau:

* **Page result**(**\**Magento\Framework\View\Result\PageFactory): Loại thông dụng nhất, trả về 1 trang html (được tạo ra bởi layout, block và template).
* **JSON result**(\Magento\Framework\Controller\Result\JsonFactory): Trả về 1 JSON object, dùng khi muốn thiết kế các APIs (dùng trong các web service hoặc trả về kết quả cho views sử dụng AJAX)
* **Raw result**(\Magento\Framework\Controller\Result): Trả về 1 plain string (không có layout hay template)
* **Forward result**(\Magento\Framework\Controller\Result\ForwardFactory): Điều hướng sang 1 **action** khác nhưng không gửi lại 1 HTTP request mới (không thay đổi URL)
* **Redirect result**(\Magento\Framework\Controller\Result\RedirectFactory): Điều hướng sang 1 **route** khác bằng HTTP request (tương đương với việc người dùng truy cập vào 1 route khác, URL thay đổi)

*Source code ví dụ mẫu* <https://github.com/robinhuy/magento2-example/tree/v1.1>

## Bài 4

Tìm hiểu thêm về RequireJS tại <http://requirejs.org/>

**Chú ý**

- Khi khai báo đường dẫn trong **requirejs-config** thì không có đuôi **.js** ở cuối. Ví dụ với file myscript.js nằm trong thư mục**web/js**thì khai báo đường dẫn là **Vendor\_Module/js/myscript**

- Trong 1 số trường hợp nếu bị lỗi **requirejs**thì có thể thử xóa thư mục **pub/static/\_requirejs** đi và load lại module.

- Trong magento có sẵn một số thư viện jQuery dùng để tạo các widget, chỉ cần require vào là sử dụng được không cần khai báo thêm trong requirejs-config: <http://devdocs.magento.com/guides/v2.1/javascript-dev-guide/widgets/jquery-widgets-about.html>. Ví dụ sử dụng [Calendar Widget](http://devdocs.magento.com/guides/v2.1/javascript-dev-guide/widgets/widget_calendar.html):

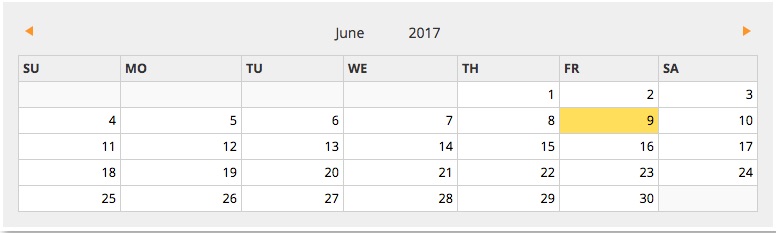
require(['jquery', 'mage/calendar'], **function** ($) {

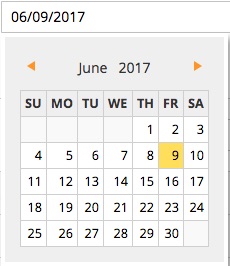
$(**function** () {

$("#datepicker").datepicker();

}

})

Tạo Calendar bằng thẻ <div id="datepicker"></div>

Tạo Datepicker bằng thẻ <input type="text" id="datepicker"/>

**Luyện tập**

Trình bày lại giao diện module **Dự báo thời tiết** với giao diện đẹp hơn và có thông số dự báo thời tiết cho cả tuần tiếp theo (forecast).

*Source code ví dụ mẫu* <https://github.com/robinhuy/magento2-example/tree/v1.2>

## Bai 5

* [Bảng xếp hạng](https://techmaster.vn/khoa-hoc-online/25520/xay-dung-website-ban-hang-voi-magento-2/3561/Setup-scripts#tab_student_list)

File đính kèm

* [Huong dan cai dat phpMyadmin tren Ubuntu.docx](https://video1.techmaster.vn/stream/attach_files/download/1076/jgD0DM23nLCBMM1BP2NJbVXobDu2esAgw2pu8TulXbDW04Ul7g/eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJwcmljZSI6MiwiZnVsbCI6dHJ1ZSwidmlkZW8iOnsiaWQiOjExNTAsIm5hbWUiOiJtYWdlbnRvLXNldHVwLXNjcmlwdHMubXA0IiwicGF0aCI6Ii9tZWRpYS8yNTUyMCJ9LCJhdHRhY2htZW50IjpbeyJpZCI6MTA3NiwibmFtZSI6Ikh1b25nIGRhbiBjYWkgZGF0IHBocE15YWRtaW4gdHJlbiBVYnVudHUuZG9jeCIsInBhdGgiOiIvbWVkaWEvMjU1MjAifV19.q717n95PMZvZQUEC2fhWl_OPkOiuPu3pDXNetJ3bIus)

**Chú ý**

- Bảng setup\_modules trong database sẽ lưu trữ thông tin về các module đã cài đặt kèm theo version của chúng.

- Ngoài **InstallSchema** script, còn có các script khác như:

* **UpgradeSchema**: Thay đổi cấu trúc bảng khi module được Update (khi module ra version mới, có thể test bằng cách tăng version của module sau đó chạy lệnh **bin/magento** **setup:upgrade**)
* **InstallData** + **UpgradeData**: Insert dữ liệu, cập nhật dữ liệu cho bảng, chạy sau các script InstallSchema và UpgradeSchema.
* **Recurring**+**RecurringData**: Script chạy sau các setup scripts khác (schema và data), dùng để thực hiện các bước cuối cùng của cài đặt (ví dụ như thực hiện update index)
* **Uninstall**: Chạy khi module được uninstall, dùng để xóa hết các table, dữ liệu liên quan đến module

- Khi tạo bảng nên đặt tên bảng kèm theo prefix ví dụ tên Vendor như **robin\_banner** để tránh các module khác nhau sử dụng bảng giống nhau.

- Tham khảo thêm về Extension Lifecycle: <http://devdocs.magento.com/guides/v2.0/extension-dev-guide/prepare/lifecycle.html>

**Luyện tập**

Viết script **UpgradeSchema** có tác dụng như sau:

- Kiểm tra xem trong database có bảng **banner** chưa, nếu chưa có thì tạo bảng với các cột sau: id (int primary key), image (varchar 255), link (varchar 255), sort\_order (small int), status (boolean)

- Nếu trong database có bảng **banner** rồi thì bổ sung thêm cột **sort\_order (small int) và status (boolean)** (giả sử là ở version trước chưa có 2 cột này). Có thể dùng phương thức run($sql) của interface **SetupInterface** để chạy 1 câu lệnh raw SQL

- Bảng **banner** sẽ được sử dụng tiếp cho các ví dụ sau, dùng để tạo banner (hoặc slider) ở trang chủ của website.

*Source code ví dụ mẫu* <https://github.com/robinhuy/magento2-example/tree/v2.0>

## Bài 6

Tham khảo các phương thức tương tác với model trong class **\Magento\Framework\Model\AbstractModel** tại thư mục **lib > internal > Magento > Framework > Model > AbstractModel.php**

**Chú ý**

- Không phải Model nào cũng dùng để tương tác với Cơ sở dữ liệu

- Ngoài cách khởi tạo model bằng **objectManager** như trong video chúng ta có thể sử dụng **Factory class** (code dài hơn 1 chút nhưng tường minh hơn). 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. Tham khảo cách khởi tạo object bằng factory tại [Instantiating objects with factories](http://devdocs.magento.com/guides/v2.0/extension-dev-guide/factories.html)

Ví dụ sử dụng Factory để khởi tạo model

**protected** $bannerFactory;

**public** **function** **\_\_construct**(\Magento\Framework\App\Action\Context $context, \Robin\Bai2\Model\BannerFactory $bannerFactory)

{

$this->bannerFactory = $bannerFactory;

**parent**::\_\_construct($context);

}

**public** **function** **execute**()

{

// Create banner instance

$banner = $this->bannerFactory->create();

// Get banner with id = 1

$data = $banner->load(1)->getData();

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

}

**Code mẫu khai báo Model**

// File Model/Banner.php

<?php

**namespace** **Robin**\**Bai2**\**Model**;

**class** ***Banner*** **extends** \***Magento***\***Framework***\***Model***\***AbstractModel***

{

**protected** **function** **\_construct**()

{

$this->\_init('Robin\Bai2\Model\ResourceModel\Banner');

}

}

// File Model/ResourceModel/Banner.php

<?php

**namespace** **Robin**\**Bai2**\**Model**\**ResourceModel**;

**class** ***Banner*** **extends** \***Magento***\***Framework***\***Model***\***ResourceModel***\***Db***\***AbstractDb***

{

**protected** **function** **\_construct**()

{

// Table name + primary key column

$this->\_init('banner', 'id');

}

}

// File Model/ResourceModel/Banner/Collection.php

<?php

**namespace** **Robin**\**Bai2**\**Model**\**ResourceModel**\**Banner**;

**class** ***Collection*** **extends** \***Magento***\***Framework***\***Model***\***ResourceModel***\***Db***\***Collection***\***AbstractCollection***

{

**protected** **function** **\_construct**()

{

// Model + Resource Model

$this->\_init('Robin\Bai2\Model\Banner', 'Robin\Bai2\Model\ResourceModel\Banner');

}

}

**Luyện tập**

Viết controller để nhập dữ liệu cho bảng **banner** 1 cách tự động:

- Nhập **100** bản ghi bao gồm dữ liệu đầy đủ cho các cột.

- Cột **image** chứa tên ảnh phải khác nhau, phần mở rộng của ảnh là 1 trong 3 đuôi sau: **png**, **jpg**, **gif.**

- Cột **link** chứa đường link mà ảnh trỏ đến, phải có định dạng là 1 URL.

- Cột **sort\_order** lưu vị trí của ảnh từ 1 đến 100.

- Cột **status** lưu trạng thái là **true** hoặc **false**(0 hoặc 1).

*Source code ví dụ mẫu* <https://github.com/robinhuy/magento2-example/tree/v2.1>

## Bai 7

Tham khảo các phương thức tương tác với collection trong class **\Magento\Framework\Data\Collection\AbstractDb** tại thư mục **lib > internal > Magento > Framework > Data > Collection > AbstractDb.php**

Ôn lại 1 số câu lệnh Linux hay dùng: [29 câu lệnh Linux bạn cần biết phần 1](https://techmaster.vn/posts/33519/cau-lenh-linux-can-biet-1) và [29 câu lệnh Linux bạn cần biết phần](https://techmaster.vn/posts/33520/cau-lenh-linux-ban-can-biet-2)[2](http://techmaster.vn/posts/33520/cau-lenh-linux-ban-can-biet-2)

**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 |

*Source code ví dụ mẫu* <https://github.com/robinhuy/magento2-example/tree/v2.2>

## Bai 8

**Chú ý**

- Backend có thể hiểu theo nhiều nghĩa khác nhau, tuy nhiên trong bài học chúng ta sẽ hiểu theo nghĩa là trang quản trị (admin area) dùng để quản lý danh mục, sản phẩm, bài viết, ....

- Trong code của Magento 2 sẽ có các thư mục tượng trưng cho các thành phần hiển thị khác nhau gọi là **area code**, ví dụ như: adminhtml (trang quản trị), frontend (trang giao diện người dùng), webapi\_rest (web API REST), ...

**Mục tiêu của module Quản lý banner**

- Có trang quản lý ở backend cho phép admin tạo, sửa, xóa, sắp xếp các ảnh trong banner.

- Trang frontend (trang chủ) sẽ hiển thị banner dưới dạng hình ảnh (nếu có nhiều hình ảnh thì hiển thị dưới dạng slider). Người dùng bấm vào ảnh sẽ chuyển sang trang bài viết (hoặc trang) tương ứng với ảnh đó.

**Database**

-**image**: Lưu đường dẫn ảnh

-**link**: Lưu link của bài viết

-**sort\_order**: Sắp xếp vị trí hiển thị của các ảnh trong slide nếu hiển thị nhiều ảnh

-**status**: Trạng thái của ảnh (enable hoặc disable, tương ứng với 1 hoặc 0)

*Source code ví dụ mẫu* <https://github.com/robinhuy/magento2-example/tree/v3.0>

## Bai 8

**Các thuộc tính của menu item**

- **id**: Mã của menu item, chỉ chấp nhận các ký tự theo pattern sau: [A-Za-z0-9/\_:]{3,}

- **title**: Tiêu đề hiển thị của menu

- **toolTip**: Text hiển thị khi hover lên menu 1 lúc (thuộc tính **title** trong HTML)

- **module**: Tên module mà menu này phụ thuộc vào

- **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

- **action**: Đường dẫn khi click vào menu item

- **parent**: Menu cha. Giá trị là id của menu item khác

- **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**:

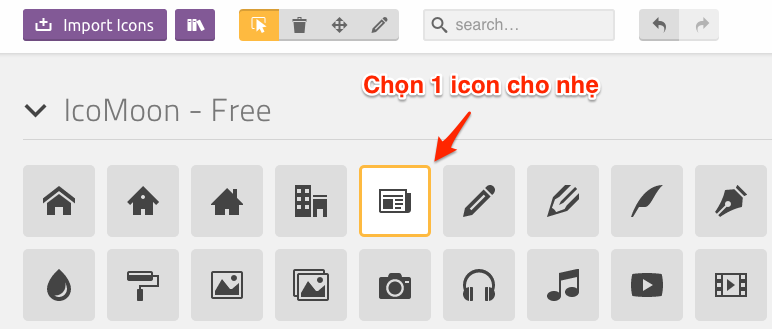
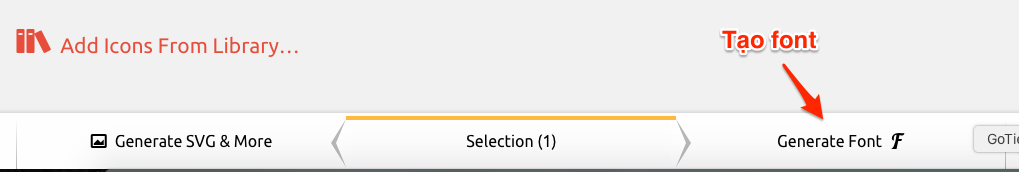
<**remove** id="Magento\_Backend::content"/>

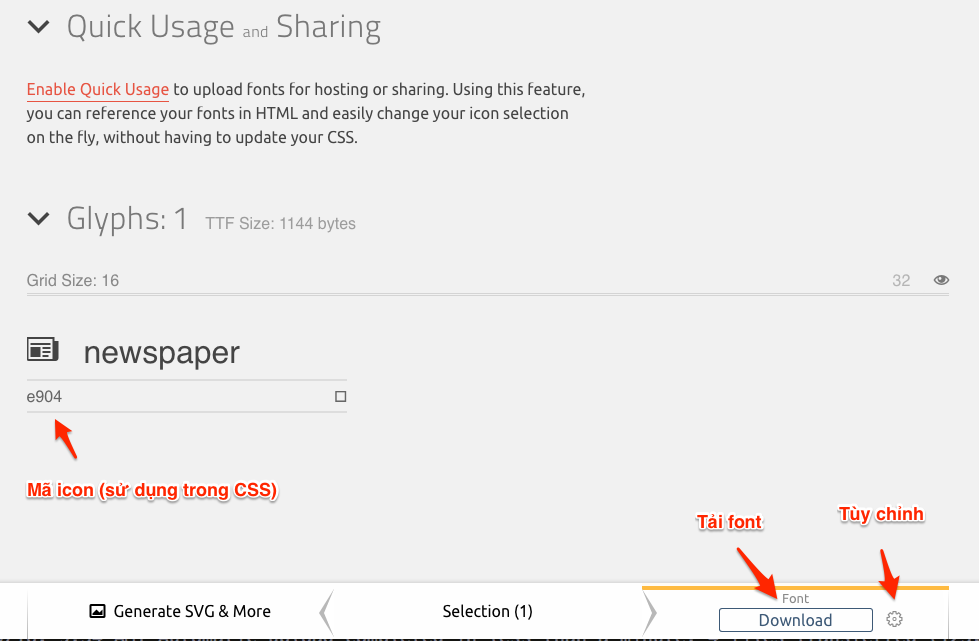
**Cách tạo custom icon cho menu**

**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

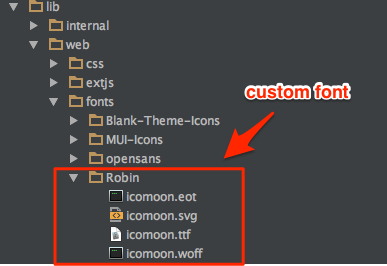
Chọn iconTạo font từ icon đã chọn

Tải font về máy

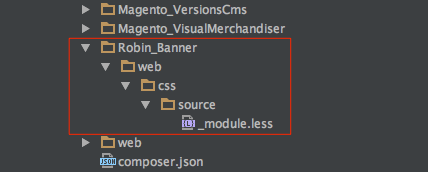
**2. Nhúng font vào source code**

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

- 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 Robin\_Banner). Sau đó tạo file \_module.less theo cấu trúc sau:



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

@**robin-icons-admin\_\_font-name-path:** '@{baseDir}fonts/Robin/icomoon'; /\* Đường dẫn đến file fonts (không có đuôi) \*/

@**robin-icons-admin\_\_font-name** : 'Robin'; /\* Font name tự đặt \*/

.lib-font-face(

@**family-name:@robin-icons-admin\_\_font-name,**

@font-path: @robin-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: @robin-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.

*Source code ví dụ mẫu* <https://github.com/robinhuy/magento2-example/tree/v3.1>

## Bai 9

Tham khảo **Dependency injection** <http://devdocs.magento.com/guides/v2.1/extension-dev-guide/depend-inj.html>

Tham khảo **Listing/Grid Component** trong UI Component <http://devdocs.magento.com/guides/v2.1/ui-components/ui-listing-grid.html>

**Một số element chính trong Grid Component**

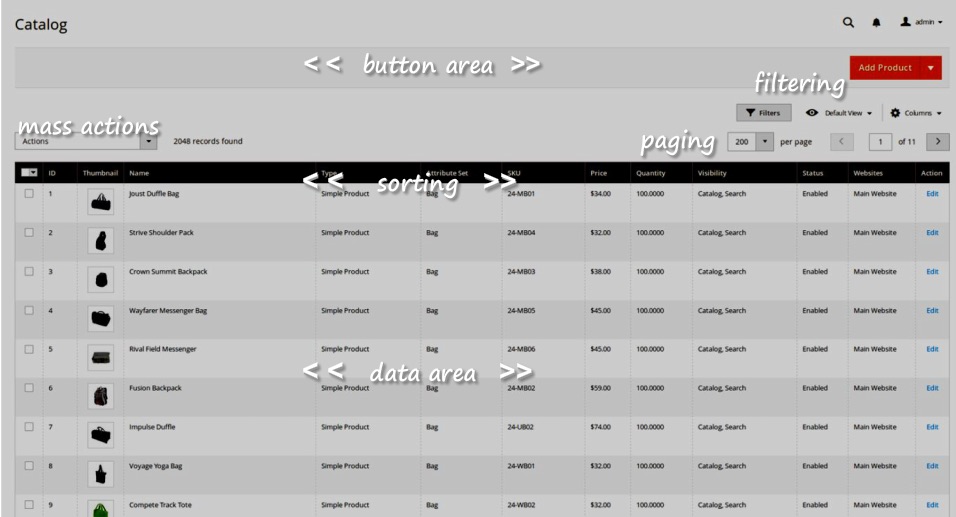
- **dataSource**: Nguồn dữ liệu đổ ra bảng

- **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">Robin\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** **Robin**\**Banner**\**Model**\**Banner**\**Source**;

**class** ***Status*** **implements** \***Magento***\***Framework***\***Data***\***OptionSourceInterface***

{

**protected** $banner;

**public** **function** **\_\_construct**(\Robin\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')];

}

**Luyện tập**

Tạo Module Banner có những chức năng sau:

- Có giao diện hiển thị danh sách dữ liệu ở backend bao gồm các chức năng: **Sorting** (sắp xếp dữ liệu theo cột), **Filter** (lọc dữ liệu). Lọc dữ liệu gồm có: Lọc theo **ID**(text range), **link**(text), **image**(text), **status**(select)

- Menu chính có icon (không dùng icon mặc định), trong đó có 2 menu con, 1 cái trỏ vào trang danh sách (ví dụ /banner), 1 cái trỏ vào trang tạo mới (ví dụ /banner/new)

- Có nút **Add New Image,**link trỏ đếntrang tạo mới

*Source code ví dụ mẫu* <https://github.com/robinhuy/magento2-example/tree/v3.2>

## Bai 10

**Luyện tập**

- Hoàn thiện các action sau: **NewAction** (trang thêm mới), **Edit** (trang chỉnh sửa) và **Delete**(trang xóa)

- Action **Delete** có thể tham khảo ở module **Cms**. Chú ý trong đó có đoạn code:

$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)

*Source code ví dụ mẫu* <https://github.com/robinhuy/magento2-example/tree/v3.3>

## Bai 11

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 **Robin\_Banner::save**

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

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

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

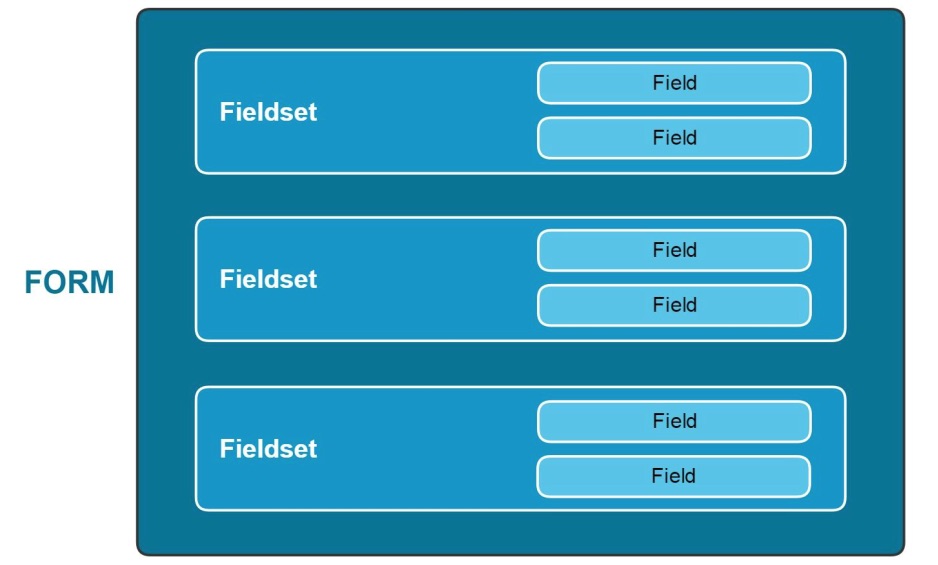
**Luyện tập**

Hoàn thành cấu hình phân quyền cho module bao gồm cả menu và các action: New, Edit, Delete.

*Source code ví dụ mẫu* <https://github.com/robinhuy/magento2-example/tree/v3.4>

## Bai 12

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**

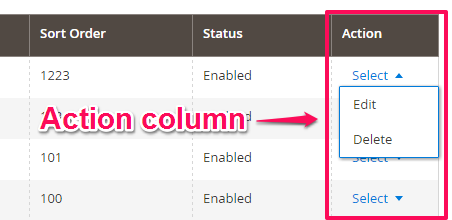
*Source code ví dụ mẫu* <https://github.com/robinhuy/magento2-example/tree/v3.5>

## Bai 13

**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="Robin\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 **Robin\Banner\Ui\Component\Listing\Column\Actions**với nội dung như sau (tham khảo module *Cms*, class *PageActions*):

<?php

**namespace** **Robin**\**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;

}

}

**Luyện tập**

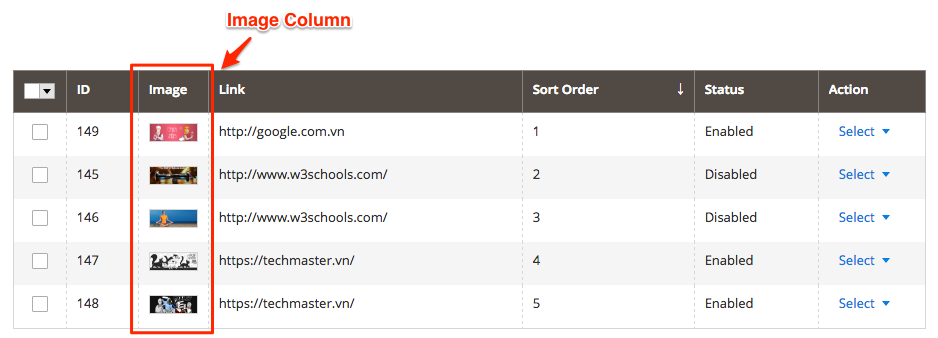
- Tạo trang **New** và **Edit**, có xử lý lưu dữ liệu vào database. Các nút trong form phải hoạt động tốt, có validate **require** với các trường bắt buộc, validate **URL** với trường *link*.

- Sửa cột Action column trang **List**để người dùng thao tác sửa, xóa dễ dàng hơn.

*Source code ví dụ mẫu* <https://github.com/robinhuy/magento2-example/tree/v3.6>

## Bai 14

**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="Robin\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 **Robin\Banner\Ui\Component\Listing\Column\Thumbnail**với nội dung như sau (tham khảo module *Catalog,*class*Thumbnail*):

<?php

**namespace** **Robin**\**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;

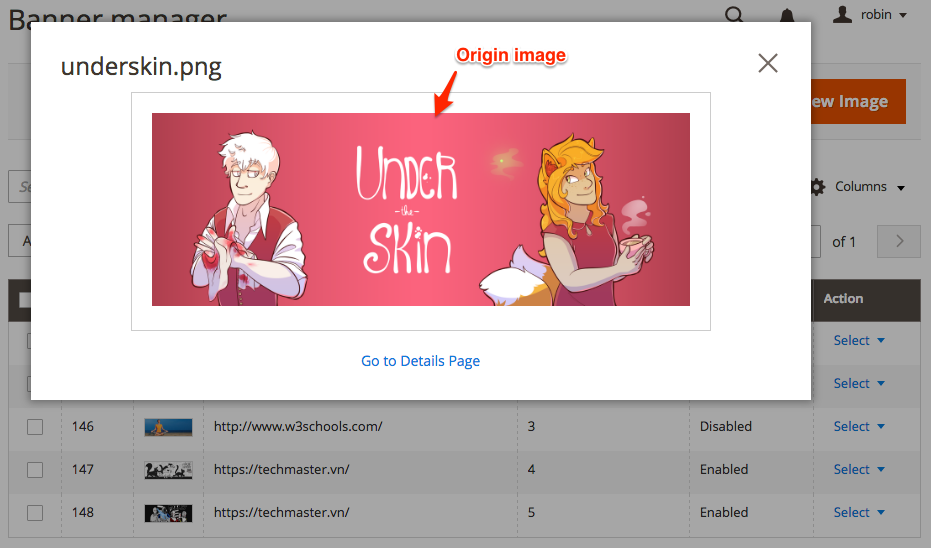
}

}

**return** $dataSource;

}

}



**Luyện tập**

- Sửa lại form **New** và **Edit** cho phép người dùng upload ảnh.

- Fix bug khi người dùng chỉnh sửa thông tin bản ghi mà không thay đổi ảnh (khi đó sẽ bị lỗi ở hàm **\_afterSave()** của model vì không có ảnh do người dùng upload lên). Gợi ý: để lấy dữ liệu mới thì dùng $object->getData(), để lấy dữ liệu cũ thì dùng $object->getOrigData()

- Ngoài trang **List** hiển thị hình ảnh thumbnail thay cho link ảnh

*Source code ví dụ mẫu* <https://github.com/robinhuy/magento2-example/tree/v3.7>

## Bai 15

**Hướng dẫn chức năng Mass Delete (xóa 1 lúc nhiều bản ghi)**

- 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** **Robin**\**Banner**\**Controller**\**Adminhtml**\**Index**;

**use** **Magento**\**Framework**\**Controller**\**ResultFactory**;

**use** **Magento**\**Backend**\**App**\**Action**\**Context**;

**use** **Magento**\**Ui**\**Component**\**MassAction**\**Filter**;

**use** **Robin**\**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** **Robin**\**Banner**\**Model**\**ResourceModel**\**Banner**;

**class** **Collection** **extends** \**Magento**\**Framework**\**Model**\**ResourceModel**\**Db**\**Collection**\***AbstractCollection***

{

**protected** $\_idFieldName = 'id';

**protected** **function** **\_construct**()

{

// Model + Resource Model

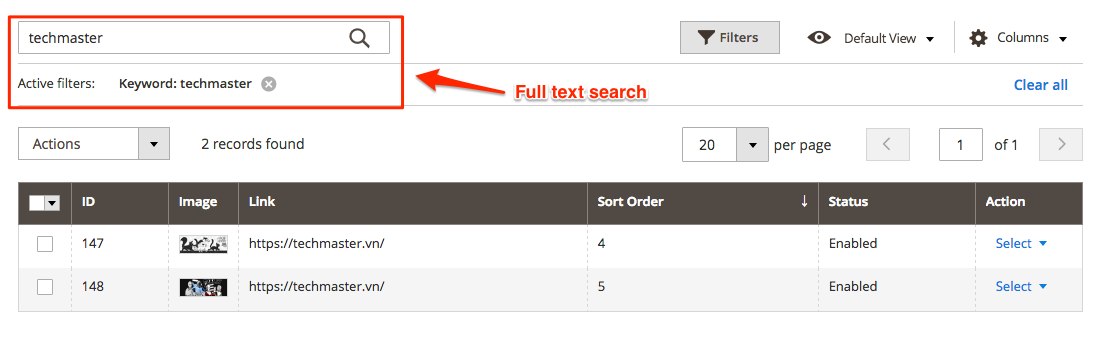
$this->\_init('Robin\Banner\Model\Banner', 'Robin\Banner\Model\ResourceModel\Banner');

}

}

- 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**

**Luyện tập**

Bổ sung chức năng **inline edit** và **mass action** cho trang **List** (mass delete, mass disable, mass enable)

*Source code ví dụ mẫu* <https://github.com/robinhuy/magento2-example/tree/v3.8>

## Bài 16

**Luyện tập**

Hoàn thiện module Quản lý Banner, cho phép người dùng tạo widget hiển thị ngoài frontend.

- Nếu không có Image nào trong bảng banner là Enable thì widget sẽ không hiển thị gì.

- Nếu chỉ có 1 Image đã Enable thì hiển thị Image kèm theo link của nó.

- Nếu có nhiều Image Enable thì hiển thị tối đa 10 Image dưới dạng Slide ảnh. Slide ảnh sẽ có các options giống với options của Wdiget cho phép người dùng tùy chỉnh như: Auto, speed, pager, .... Để tạo slide có thể dùng jQuery plugin như [bxslider](http://bxslider.com/), [owlcarousel](http://owlgraphic.com/owlcarousel/), [flexslider](https://github.com/woothemes/FlexSlider), ...

*Source code ví dụ mẫu* <https://github.com/robinhuy/magento2-example/tree/v3.9>