# \*Module

-Bước 1: Tạo folder chứa module

+Tạo 1 folder “code” bên trong folder “app”

+Tạo 1 folder chứa module (vendor) bên trong folder “app”

+Tạo 1 folder bên trong vendor, đây chính là 1 module. Tên của module được định nghĩa là “VendorName\_ModuleName”

VD: app/code/Mageplaza/HelloWorld

-Bước 2: Tạo tệp etc/module.xml

+Tạo folder etc với 1 file module.xml

VD: app/code/Mageplaza/HelloWorld/etc/module.xml

+Nội dung bên trong file module.xml như sau:

<?xml version="1.0"?>

<config xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="urn:magento:framework:Module/etc/module.xsd">

<module name="Mageplaza\_HelloWorld" setup\_version="1.0.0"> #Mageplaza\_HelloWorld là tên của module

</module>

</config>

-Bước 3: Tạo file registration.php

+Tạo 1 file registration.php bên trong module

VD: app/code/Mageplaza/HelloWorld/registration.php

+Nội dung của file sẽ là:

<?php

\Magento\Framework\Component\ComponentRegistrar::register(

\Magento\Framework\Component\ComponentRegistrar::MODULE,

'Mageplaza\_HelloWorld', #Mageplaza\_HelloWorld là tên module

\_\_DIR\_\_

);

-Bước 4: Kích hoạt module

+Kích hoạt module bằng lệnh: “sudo bin/magento module:enable Mageplaza\_HelloWorld” với Mageplaza\_HelloWorld là tên module hoặc mở file app/etc/config.php và đổi giá trị số tương ứng với module từ 0 thành 1 (không khuyến khích dùng cách này)

+Chạy lệnh “sudo bin/magento setup:upgrade” để cập nhật

# \*Controller:

-Controller là 1 phần quan trọng của quá trình phát triển module và PHP MVC framework nói chung. Chức năng của nó là tiếp nhận request, xử lý và render trang

-Trong magento2, Controller của 1 mdoule sẽ có 1 hoặc nhiều files, nó bao gồm các action của class và chứa các phương thức execute(). Có 2 loại controller khác nhau: frontedn controller và backend controller. Về cơ bản, chúng hoạt động giống nhau nhưng admin controller hơi khác 1 chút. Có 1 phương thức kiểm tra quyền trong admin controller được gọi là “form key”

-Các controller hoạt động: nó nhận 1 requets từ end-user

VD: <http://example.com/route_name/controller/action>

Với route\_name là 1 tên duy nhất được set bên trong file routes.xml của module; controller là folder bên trong Controller folder; action là 1 class với phương thức execute() để xử lý request

-Cách tạo 1 controller: Để tạo 1 controller, ta cần tạo 1 folder bên trong Controller folder của module và khai báo 1 action bên trong nó. Ví dụ, tạo 1 index controller và 1 index action cho module Mageplaza\_HelloWorld

+Bước 1: Tạo file routes.xml

Tạo 1 file app/code/Mageplaza/HelloWorld/etc/frontend/routes.xml

Với nội dung như sau:

<?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="standard">

<route frontName="helloworld" id="helloworld">

<module name="Mageplaza\_HelloWorld"/>

</route>

</router>

</config>

+Bươc 2: tạo file controller

Tạo 1 file app/code/Mageplaza/HelloWorld/Controller/Index/Index.php

Với nội dung như sau:

<?php

namespace Mageplaza\HelloWorld\Controller\Index;

class Index extends \Magento\Framework\App\Action\Action

{

protected $\_pageFactory;

public function \_\_construct(

\Magento\Framework\App\Action\Context $context,

\Magento\Framework\View\Result\PageFactory $pageFactory)

{

$this->\_pageFactory = $pageFactory;

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

}

public function execute()

{

return $this->\_pageFactory->create();

}

}

Tất cả controllder phải kế thừa class \Magento\Framework\App\Action\Action với phương thức dispatch sẽ được gọi trong phương thức execute() trong action class. Trong phương thức execute(), ta sẽ viết tất cả controller logic và trả về phản hồi cho request

+Bước 3: tạo controller Layout file

Tạo 1 file app/code/Mageplaza/HelloWorld/view/frontend/layout/helloworld\_index\_index.xml

Với nội dung như sau:

<?xml version="1.0"?>

<page xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" layout="1column" xsi:noNamespaceSchemaLocation="urn:magento:framework:View/Layout/etc/page\_configuration.xsd">

<referenceContainer name="content">

<block class="Mageplaza\HelloWorld\Block\Index" name="helloworld\_index\_index" template="Mageplaza\_HelloWorld::index.phtml" />

</referenceContainer>

</page>

+Bước 4: Tạo controller Block file

Tạo 1 file app/code/Mageplaza/HelloWorld/Block/Index.php với nội dung như sau

<?php

namespace Mageplaza\HelloWorld\Block;

class Index extends \Magento\Framework\View\Element\Template

{

}

+Bước 5: Tạo controller template file

Tạo 1 file app/code/Mageplaza/HelloWorld/view/frontend/templates/index.phtml

với nội dung như sau:

<h2>Welcome to Mageplaza.com</h2>

+Bước 6: Flush magento cache

+Bước 7: Chạy test cho controller mới

Thử truy cập URL http://<yourhost.com>/helloworld/index/index hoặc http://<yourhost.com>/helloworld/

-\Magento\Framework\App\Action\Action class cung cấp 2 phương thức quan trọng là: \_forward và \_redirect

+\_forward là 1 protected function sẽ chuyển hướng request sang 1 controller/action khác. Điều này sẽ không làm thay đổi request url

VD: $this->\_forward('hello'); #điều hướng qua action hello

+\_redirect() sẽ điều hướng request sang 1 controller/action khác, đồng thời thay đổi response header và request url

VD: $this->\_redirect('\*/\*/hello');

Sau khi truy cập từ url <http://example.com/route_name/test/forward>, url sẽ tự động bị đổi thành <http://example.com/route_name/test/hello>

# \*Route

Trong magento2, request url sẽ có dạng: http://example.com/index.php/front\_name/controller/action

Trong url đó, chúng ta sẽ thấy front\_name, nó sẽ được dùng để tìm module. Router định nghĩa tên này có mỗi module được định nghĩa trong routes.xml

Khi tạo 1 request trong magento2, nó sẽ đi theo flow sau để tìm controller/action: index.php 🡪HTTP app 🡪FrontController 🡪 Routing 🡪Controller processing 🡪 etc

FrontController sẽ được gọi trong Http class để định tuyến request tìm kiếm controller/action phù hợp

-Để tạo 1 Frontend route, ta tạo 1 file routes.xml với vị trí app/code/Mageplaza/HelloWorld/etc/frontend/routes.xml và nội dung:

<?xml version="1.0" ?>

<config xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="urn:magento:framework:App/etc/routes.xsd">

<!--Use router 'standard' for frontend route-->

<router id="standard">

<!--Define a custom route with id and frontName-->

<route frontName="helloworld" id="helloworld">

<!--The module which this route match to-->

<module name="Mageplaza\_HelloWorld"/>

</route>

</router>

</config>

Theo đó, url nên là http://example.com/index.php/helloworld/controller/action

-Để tạo 1 Admin route, ta tạo file

app/code/Mageplaza/HelloWorld/etc/adminhtml/routes.xml

với nội dung như sau:

<?xml version="1.0"?>

<config xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="urn:magento:framework:App/etc/routes.xsd">

<!--Use router 'admin' for admin route -->

<router id="admin">

<!--Define a custom route with id and frontName -->

<route id="mageplaza\_helloworld" frontName="mageplaza\_helloworld">

<!--The module which this route match to-->

<module name="Mageplaza\_HelloWorld"/>

</route>

</router>

</config>

URL của trang admin có cấu trúc giống với trang frontend, nhưng tên của admin\_area sẽ được thêm vào trước route\_frontName để nhận biết đây là một admin router

-Route có thể được sử dụng để rewrite controller, thông thường các route sẽ có thuộc tính id riêng biệt. Tuy nhiên, nếu ta khai báo 2 route có cùng id thì controller action sẽ được tìm thấy ở cả 2 modules. Magento system provide thuộc tính before/after để config thứ tự sắp xếp của module (định nghĩa module nào sẽ được tìm trước). Đây chính là logic cho việc ghi đè controller, ví dụ như thường được sử dụng để ghi đè 1 số module mặc định để custome lại theo ý muốn của user