# Làm quen với magento 2

## Cài đặt magento 2

### Cài đặt magento 2

#### Magento 2.x technology stack requirements

Đối với version Magento 2.2.x sẽ yêu cầu các php version: 7.0.13, 7.1.0, 7.2.0

Từ bản Magento 2.2.10 trở đi chỉ yêu cầu php version: 7.2, 7.1. php version 7.0 không còn được hỗ trợ

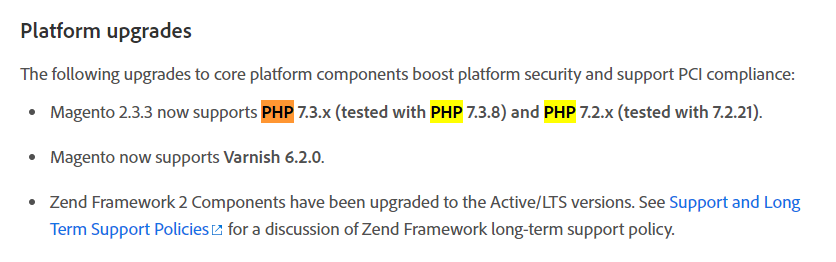
Đối với version Magento 2.3.x sẽ yêu cầu các php version: 7.1.3, 7.2.0, 7.3.0

Từ bản Magento 2.3.3 trở đi sẽ yêu cầu php version: 7.2 hoặc 7.3

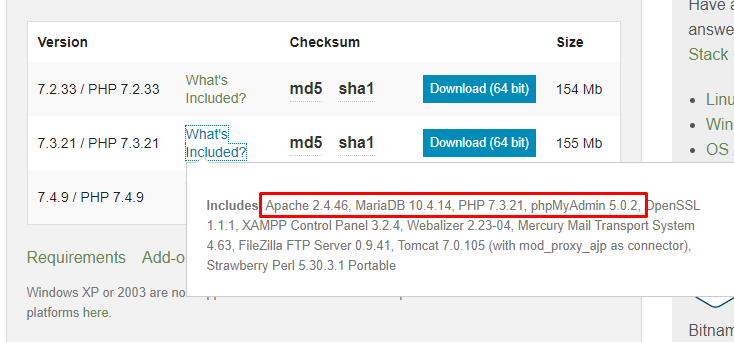
#### Intall Magento 2.3.3 với composer

1. Cài đặt Composer và Xampp với php

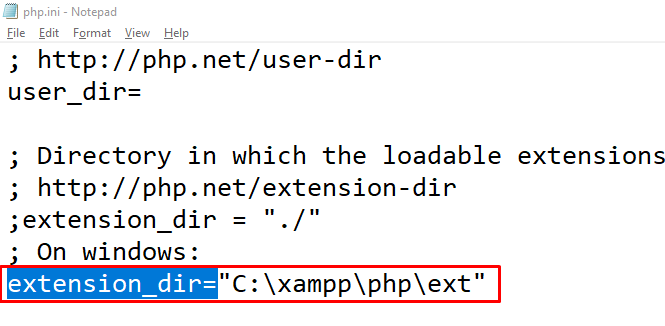
* Cài đặt composer bằng cách truy cập vào link: <https://getcomposer.org/download/> và làm theo hướng dẫn.
* Xem phiên bản php mà magento 2.2.3 hỗ trợ.



* Cài đặt xampp với PHP 7.3.x



* Mở file php.ini
* Tìm đến dòng extension\_dir= kiểm tra xem nó có bị comment ko và có trỏ đến đúng thư mục ext ko?



* Tìm đến đoạn ; Dynamic Extensions ; và bỏ ; đằng trước các extension:

extension=php\_curl.dll

extension=php\_gd2.dll

extension=php\_intl.dll

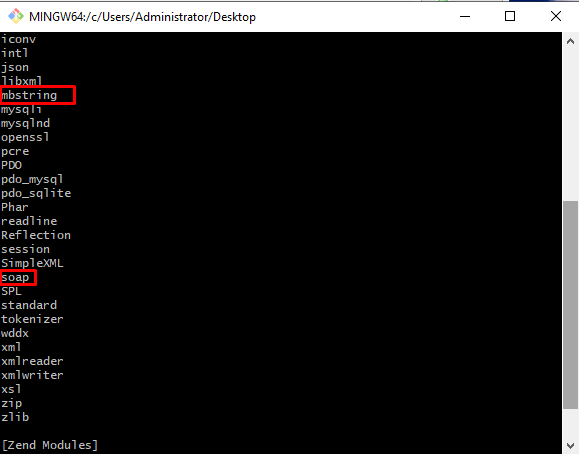
extension=php\_mbstring.dll

extension=php\_pdo\_mysql.dll

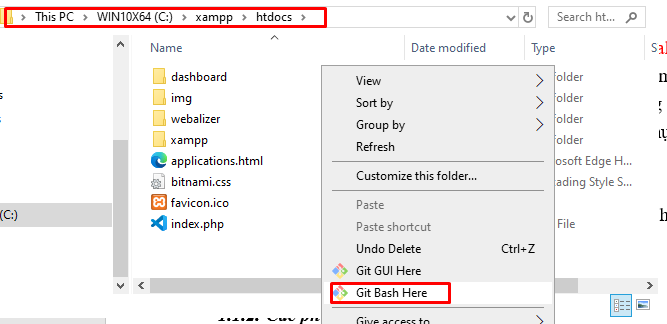
extension=php\_soap.dll

extension=php\_xsl.dll

* Sau đó dùng lệnh php –m để kiểm tra các extension được load



* Nếu gặp lỗi: 'php' is not recognized as an internal or external command, operable program or batch file. Thì vào My computer -> properties -> advance system setting enviroment variable trong tab system variable tìm dòng PATH -> edit thêm đường dẫn tới thư mục php là được , ví dụ E:\xampp7\php
* Kiểm tra version php bằng lệnh: php –v
* Vào xampp/htdocs trong xampp và chạy git bash here

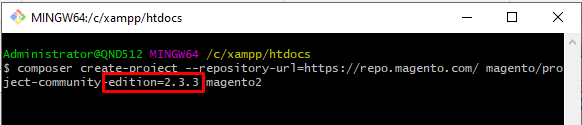


* Chạy lệnh:

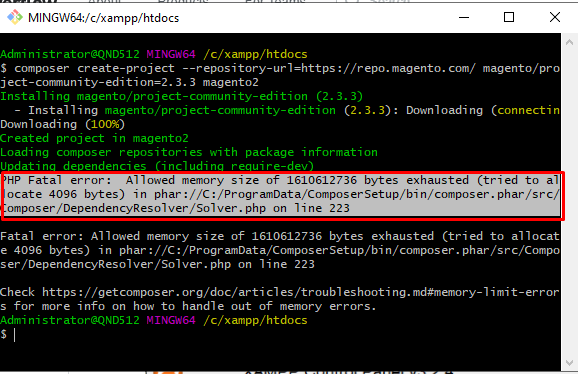
$ composer create-project --repository-url=https://repo.magento.com/ magento/project-community-edition magento2 để cài đặt phiên bản mới nhất của magento 2 hoặc

$ composer create-project --repository-url=https://repo.magento.com/ magento/project-community-edition=2.2.5 magento2 để tùy chọn version của magento 2

* Ở đây ta cài magento 2.3.3 nên chạy lệnh thứ 2

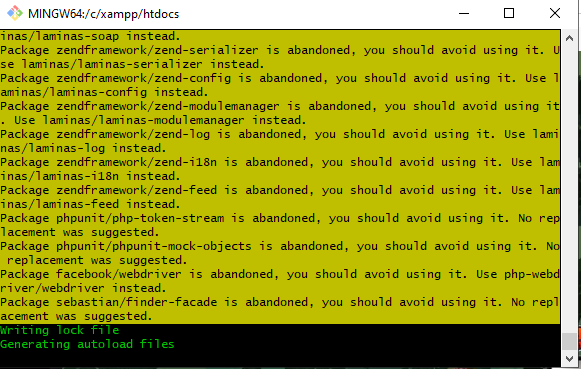


* Nếu gặp lỗi:



Thì mở file php.ini và thay đổi memory\_limit=512M thành memory\_limit=-1 xong đó xóa file magento vừa tạo trong htdocs đi.

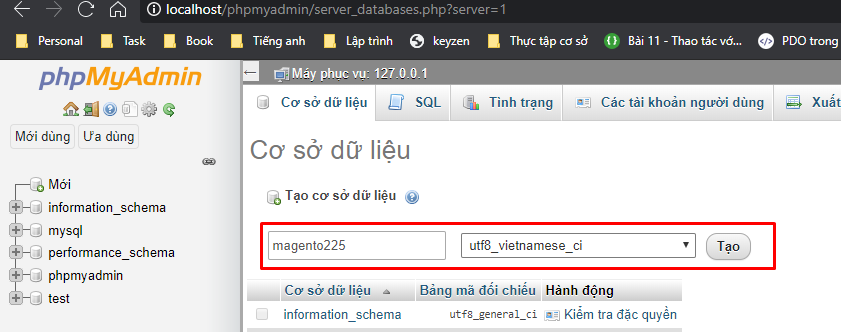
* Cài đặt magento 2 thành công máy sẽ thông báo:



* Kiểm tra version magento 2 vừa cài đặt bằng cách vào file magento2 vừa cài đặt bật git bash here gõ lệnh: php bin/magento –version



* Sau khi cài magento 2 thành công thì ta phải tạo và cập nhật database bằng cách vào phpmyadmin tạo một database



* Tạo các tables, data bằng câu lệnh trong git bash here:

php bin/magento setup:install \

--base-url=http://localhost/magento2/ \

--db-host=localhost \

--db-name=magento225 \

--db-user=root \

--admin-firstname=Magento \

--admin-lastname=User \

--admin-email=user@example.com \

--admin-user=admin \

--admin-password=admin123 \

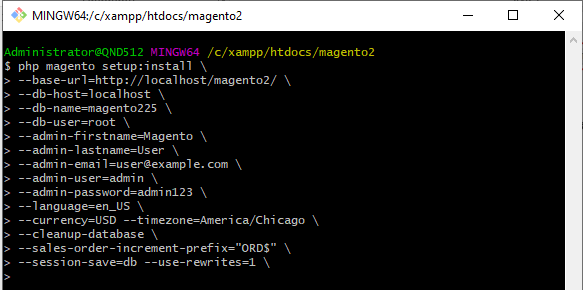
--language=en\_US \

--currency=USD --timezone=America/Chicago \

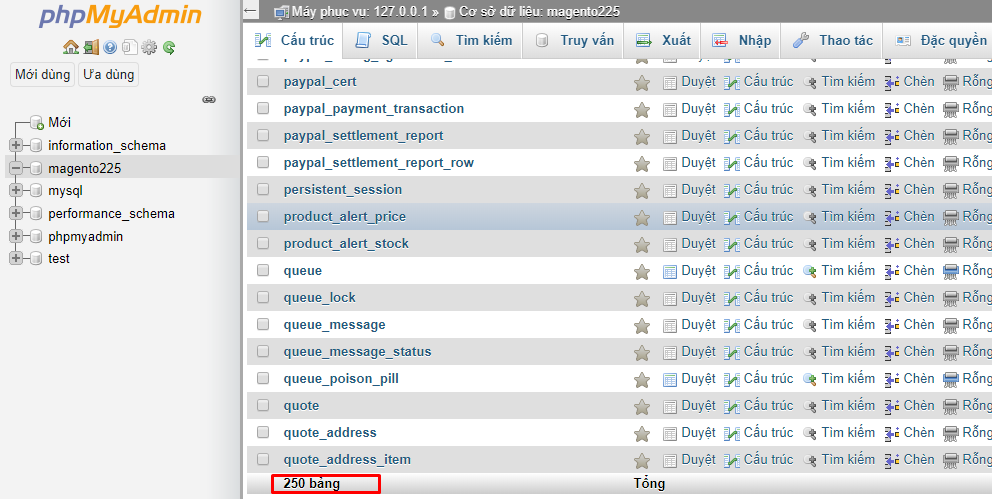
--cleanup-database \

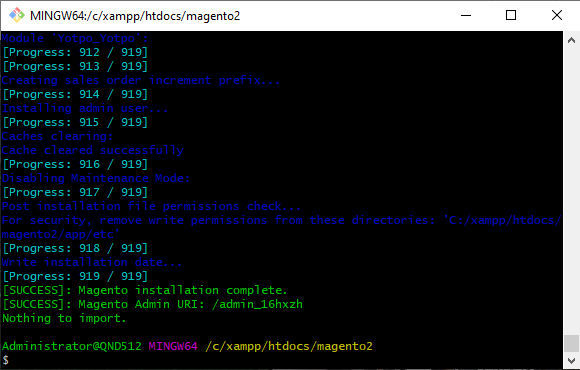
--sales-order-increment-prefix="ORD$" \

--session-save=db --use-rewrites=1 \

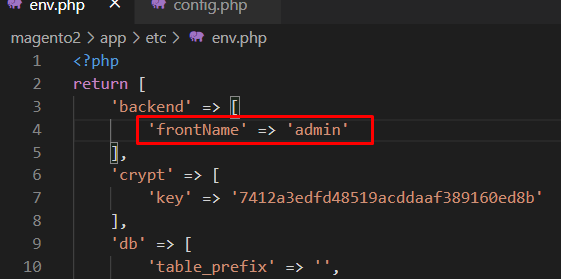


* Khi cài đặt xong ta có thông báo Admin URI là đường dẫn tới trang admin của magento 2

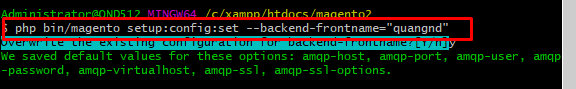




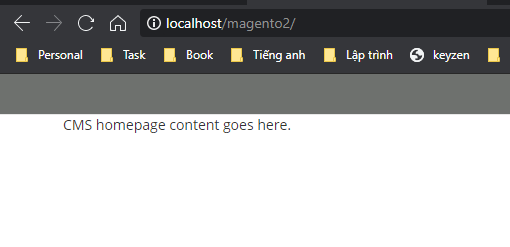
* Ở đây URL admin: localhost/magento2/admin\_16hxzh
* Để thay đổi admin\_16hxzh thành admin ta vào /app/etc/env.php để thay đổi



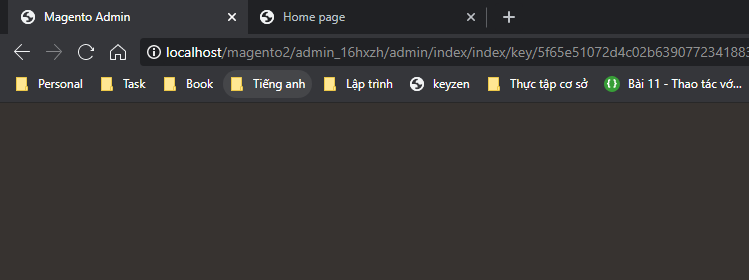
Hoặc cũng có thể dùng câu lệnh: php bin/magento setup:config:set --backend-frontname=”adminname”



* Khi truy cập vào localhost/magento2 thì thấy nó bị lỗi



* Và không thể truy cập được vào admin



* Ta xử lý lỗi sau bằng cách vào: \vendor\magento\framework\View\Element\Template\File\Validator.php tìm dòng code protected function isPathInDirectories($path, $directories) và thay thế bằng:

protected function isPathInDirectories($path, $directories) {

if (!is\_array($directories)) {

$directories = (array)$directories;

}

$realPath = $this->fileDriver->getRealPath($path);

$realPath = str\_replace('\\', '/', $realPath); // extra code added

foreach ($directories as $directory) {

if (0 === strpos($realPath, $directory)) {

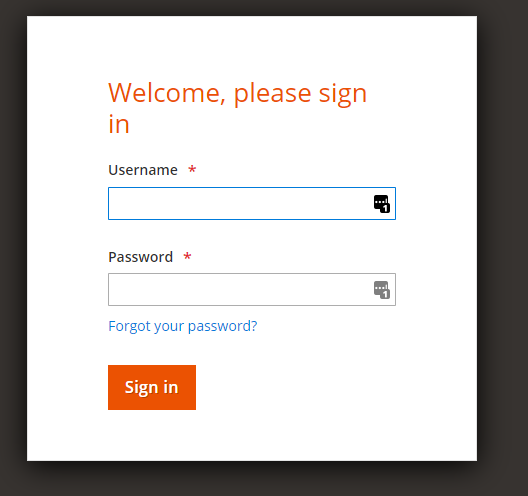
return true;

}

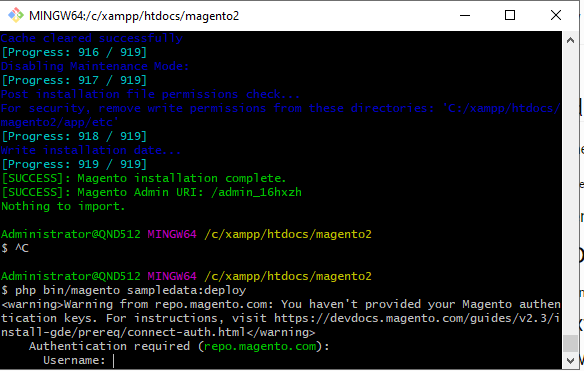
}

return false; }

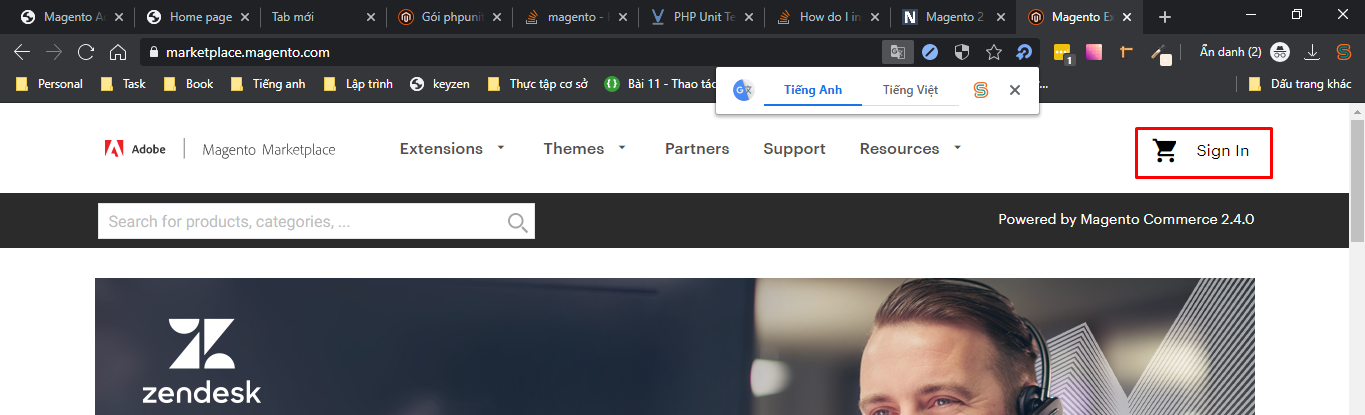
* Vào lại url tới admin và kết quả:

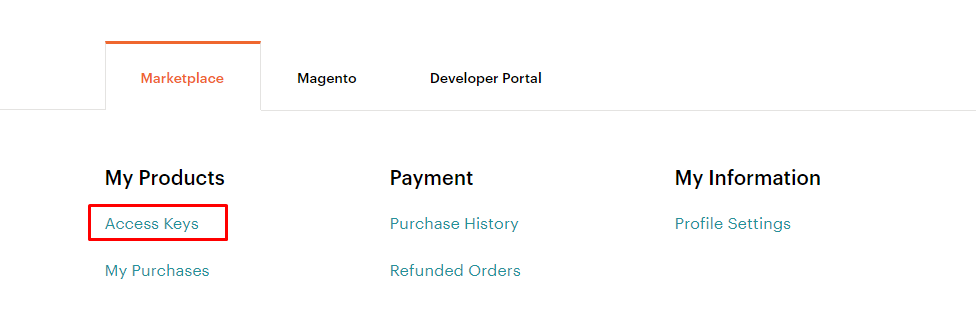


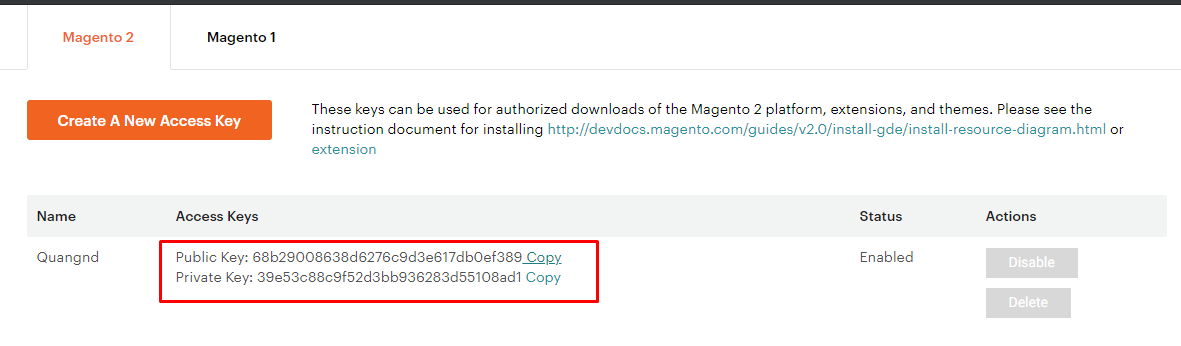
* Dùng lệnh: php bin/magento sampledata:deploy để tải về dữ liệu mẫu của magento 2

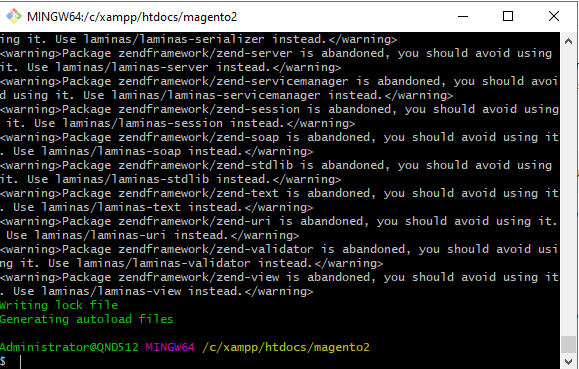


* Truy cập vào repo.magento.com đăng ký tài khoản và mật khẩu để lấy mã magento 2 cho source

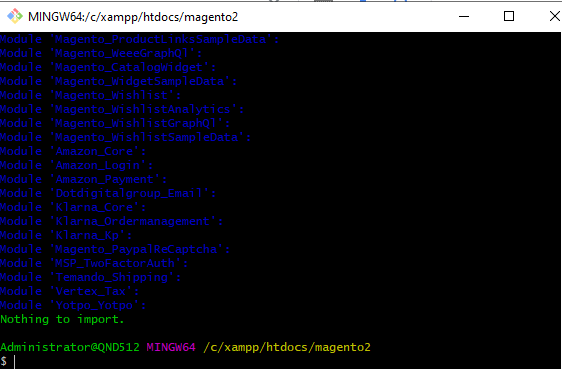




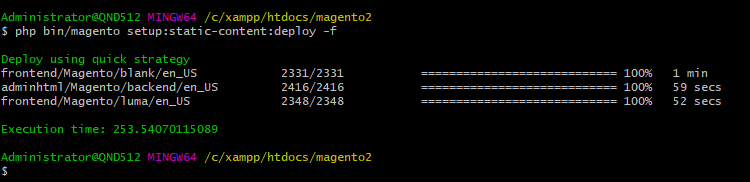




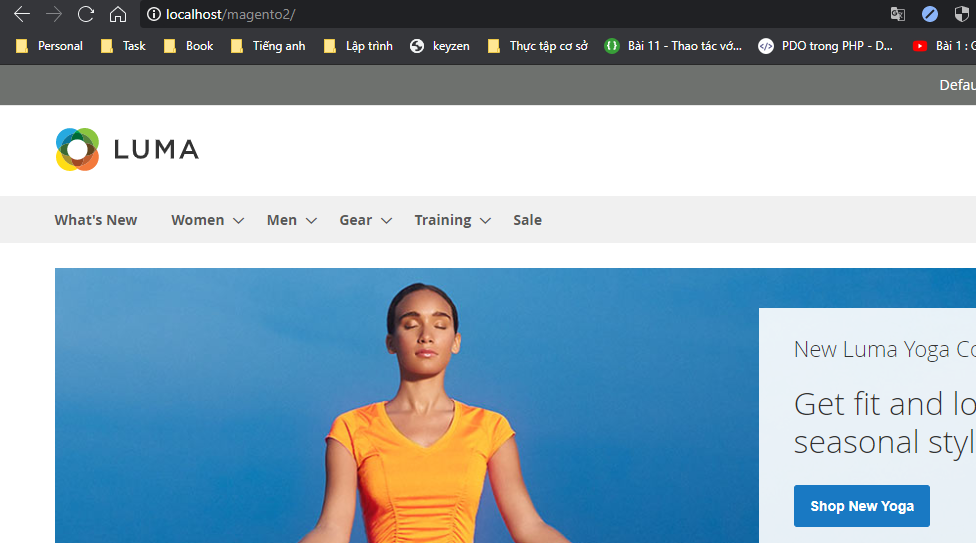
* Dùng lệnh: php bin/magento setup:upgrade để nâng cấp magento code, database.



* Dùng lệnh: php magento setup:static-content:deploy –f để chạy file trên tất cả các mode



Kết quả:



Tạo user và password cho người quản trị mới:

php bin/magento admin:user:create

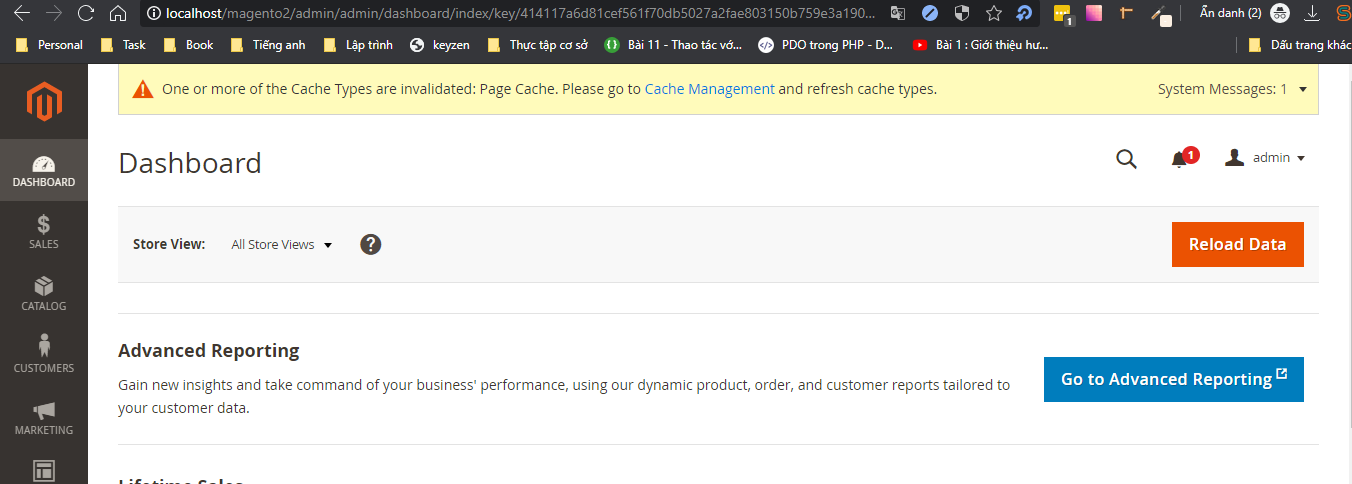
--admin-user="admin"

--admin-password="admin123@"

--admin-email="admin@gmail.com"

--admin-firstname="Admin"

--admin-lastname="Admin"



#### Thay đổi url base magento 2

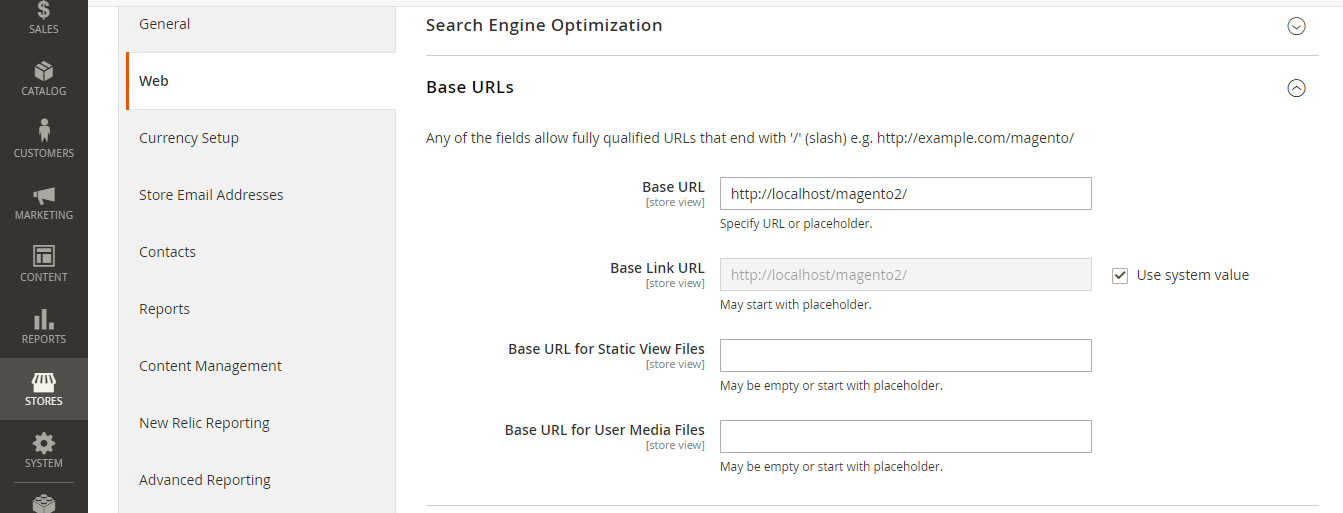
* Để thay đổi config URL website ta có 2 cách:

Cách 1: Thay đổi trong admin

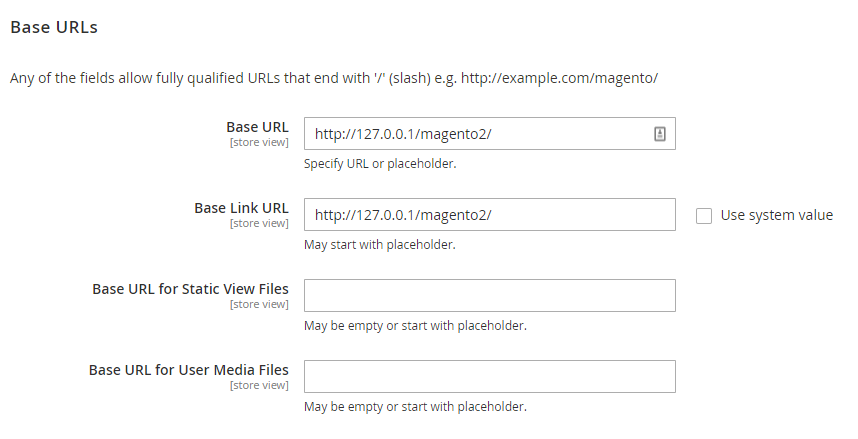
+ Trong admin vào : Stores > Settings > Configuration

+ Trong bảng điều khiển chọn: General > Web

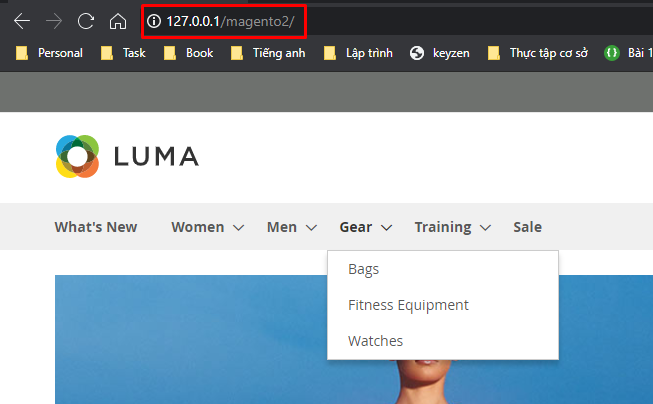
+ Để thay đổi URL website vào mục Base URLs

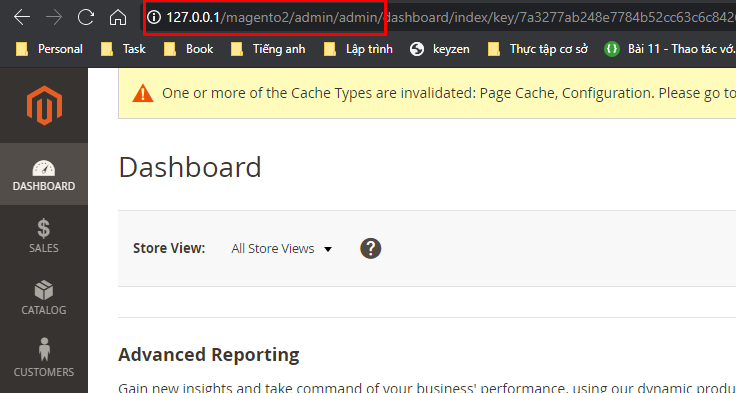


+ Sửa thành:

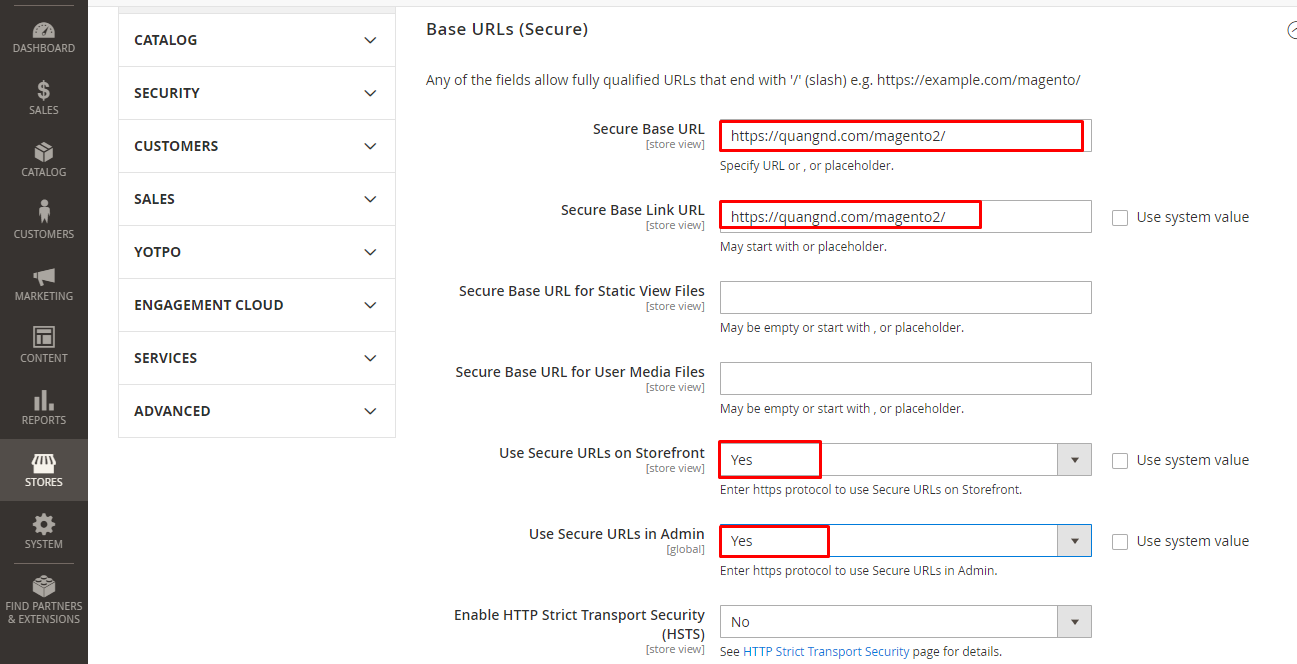


* Sau khi sửa đổi đường dẫn



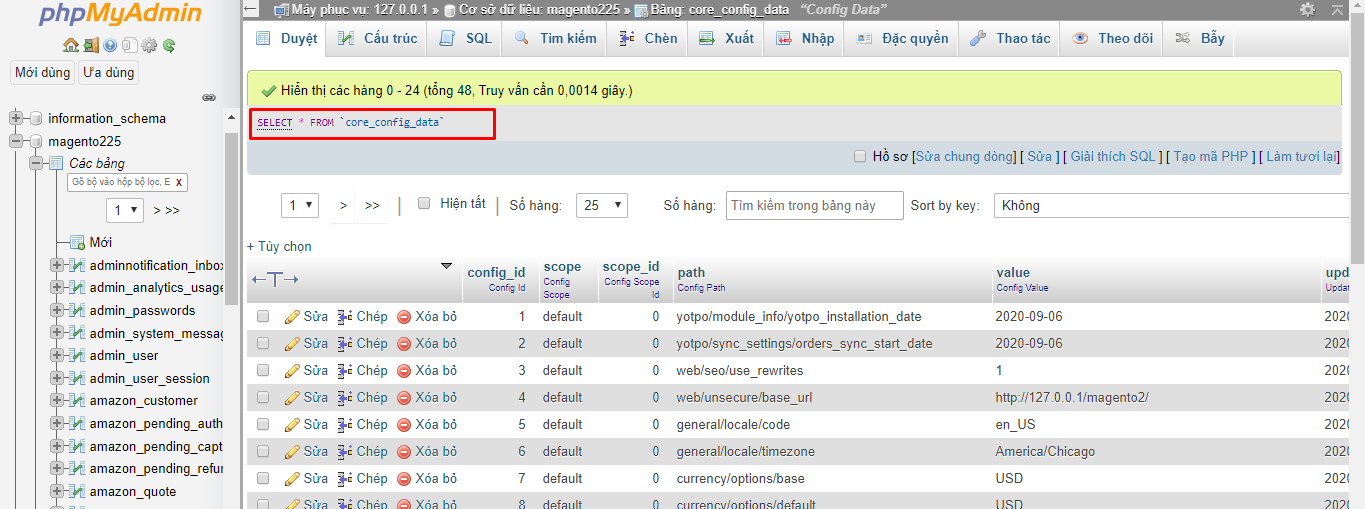


* Để sửa đổi đường dẫn thành https ta sẽ sửa trong Base URLs (Secure)

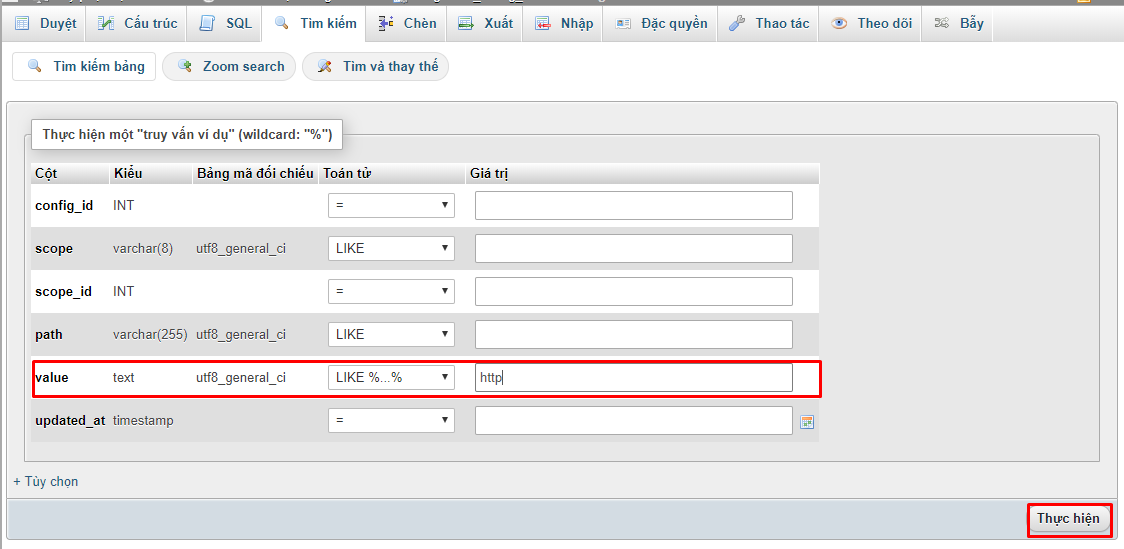


Cách 2: Thay đổi trong database

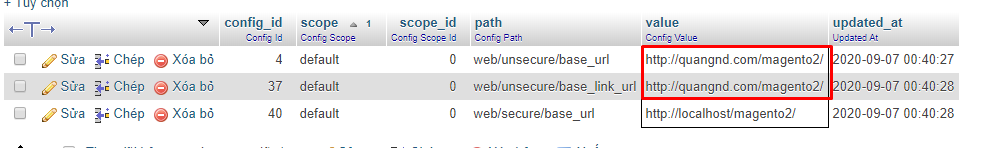
+ Vào *http://localhost/phpmyadmin* và truy cập vào CSDL của magento 2 search core\_config\_data để tìm đến bảng core\_config\_data



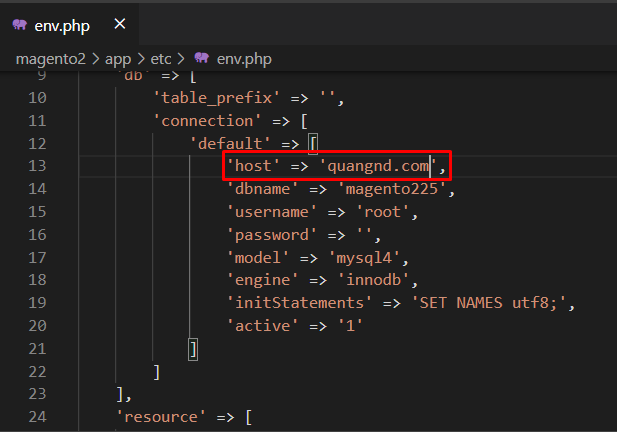
+ Sau đó nhấn vào Tìm kiếm, Ở trường value trong mục tìm kiếm. cột toán tử để là LIKE %...%. Cột giá trị điền là http sau đó nhấn thực hiện



+ Thay đổi đường dẫn mong muốn tại cột value. Ở đây ta đổi 127.0.0.1 thành quangnd.com

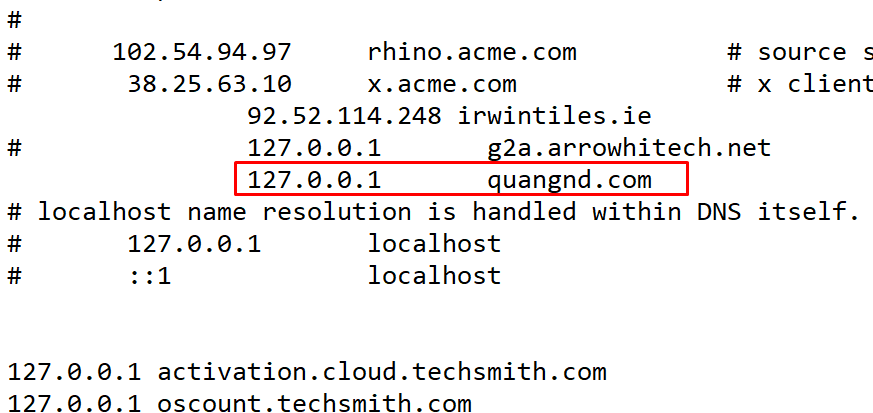


+ Và thay đổi host tại app/etc/env.php thành quangnd.com

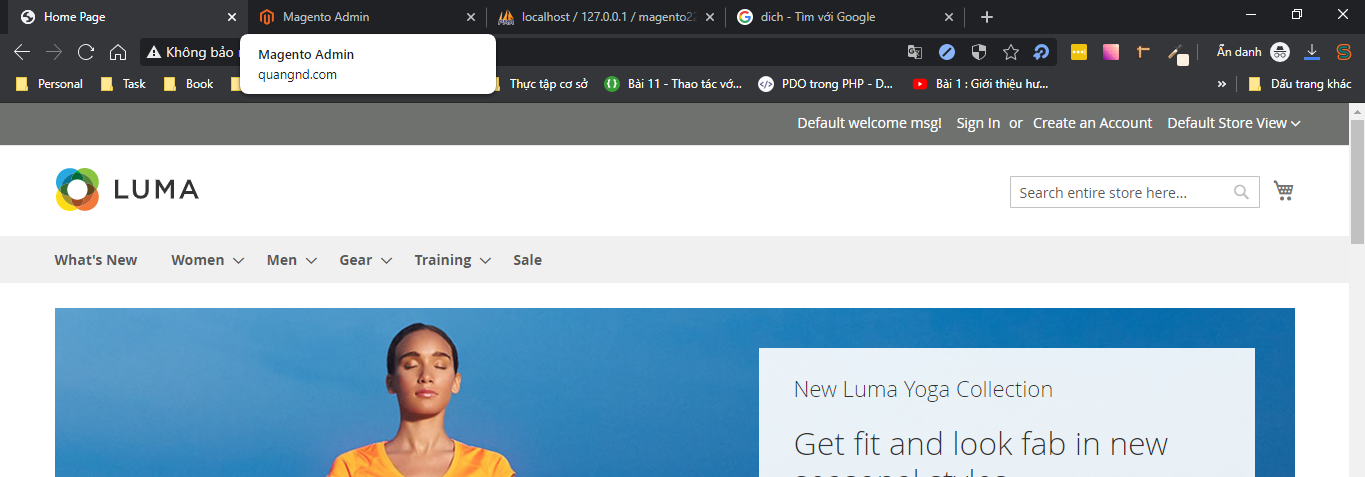


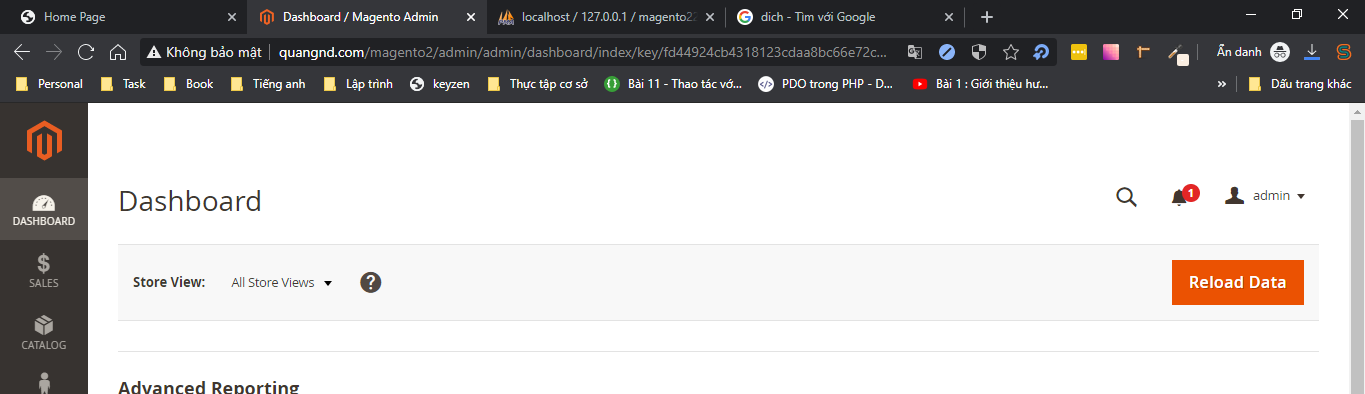
+ Vào file hosts trong C:\Windows\System32\drivers\etc thêm dòng

127.0.0.1 quangnd.com



+ Khởi động lại apache



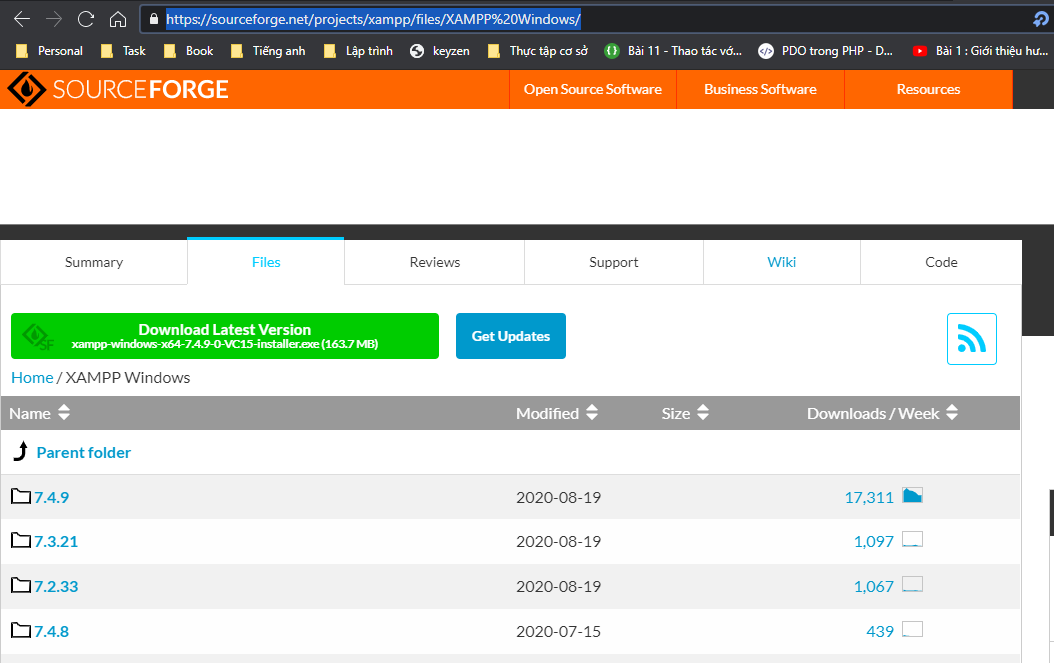


### Các phiên bản magento 2 và PHP

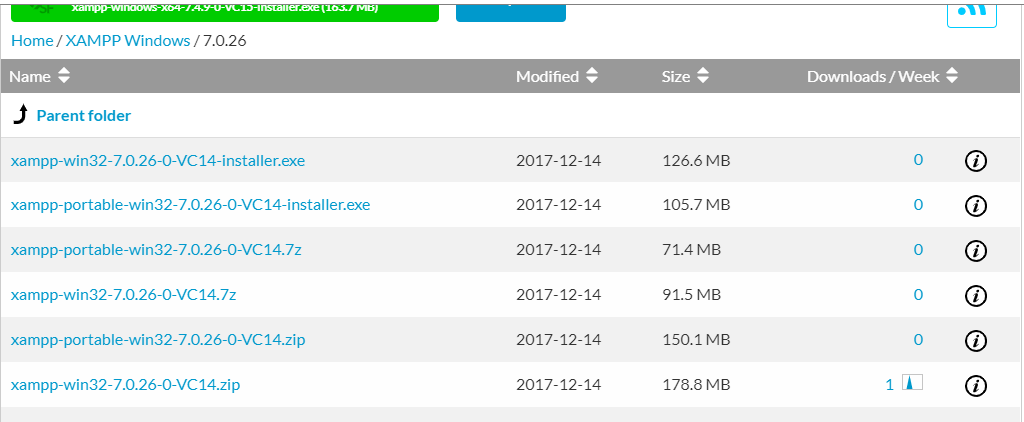
#### Lấy source magento từ github về

* Khi lấy source magento 2 trên github ta sẽ bỏ source vào xampp/htdocs. Xong đó vào trong thư mục source vừa lấy về bật git bash here gõ lệnh: composer install và cài đặt database là xong.
* Nếu bị lỗi server error thì vào thư mục magento vừa copy xóa file .htaccess đi rồi truy cập lại.
* Khi cài đặt nếu gặp lỗi: Package phpunit/phpunit-mock-objects is abandoned, you should avoid using it. No replacement was suggeste ta dùng lệnh: composer install --no-dev
* Khi gặp lỗi: Your requirements could not be resolved to an installable set of packages thì có nghĩa là phiên bản php của bạn không phù hợp với magento hiện tại cho nên bạn cần hạ cấp hoặc nâng cấp phiên bản php
* Để hạ cấp và nâng cấp phiên bản php ta làm như sau:

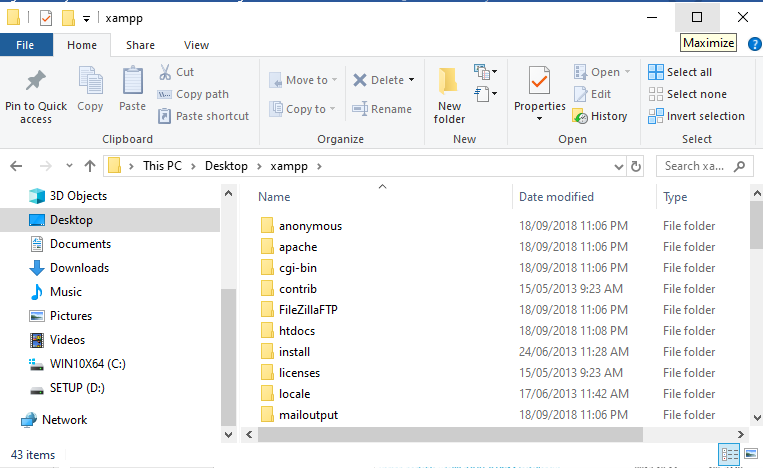
1. Dowload phiên bản php cần thiết tại <https://sourceforge.net/projects/xampp/files/XAMPP%20Windows/>



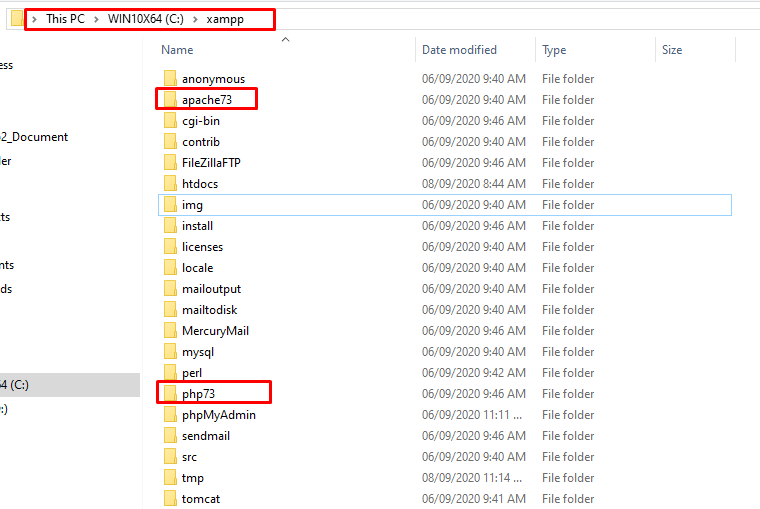
* Chọn một phiên bản xampp tương ứng và vào trong thư mục đó



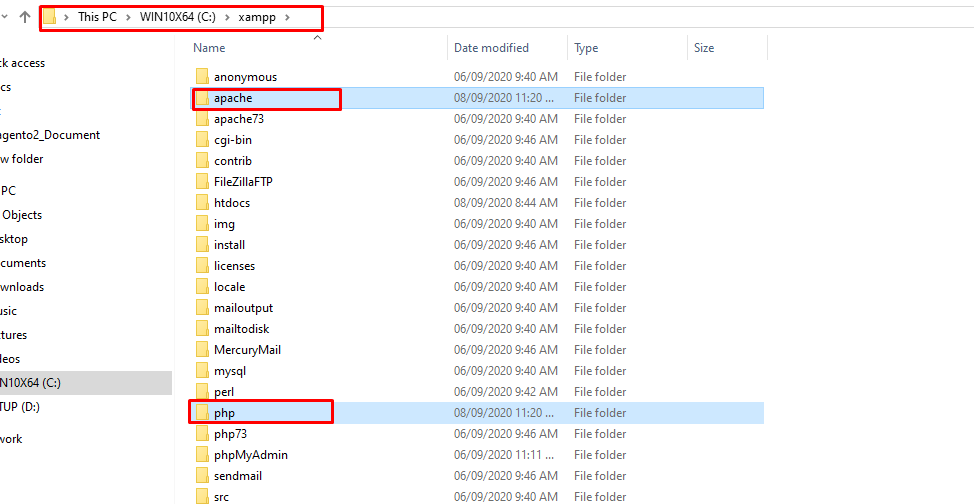
* Dowload một file .zip và tiến hành giải nén nó.



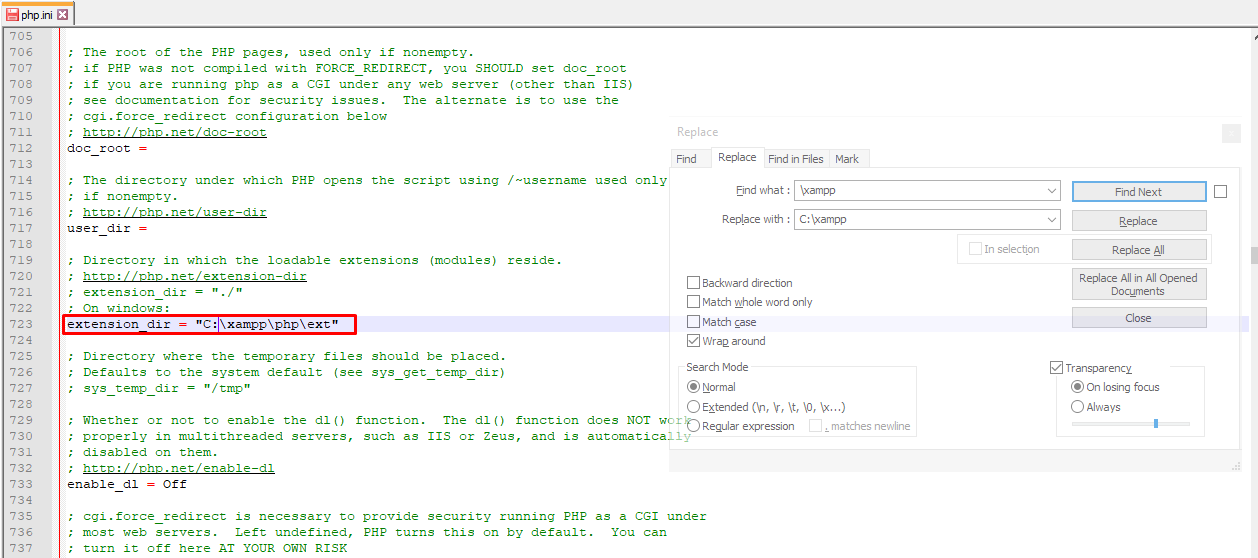
1. Đổi tên file php và file apache trong thư mục C:\xampp\ thành tên khác



1. Copy file php và file apache trong thư mục vừa tải về vào thư mục C:\xampp\



1. Vào file php mới copy và sửa php.ini dòng \xampp thành C:\xampp

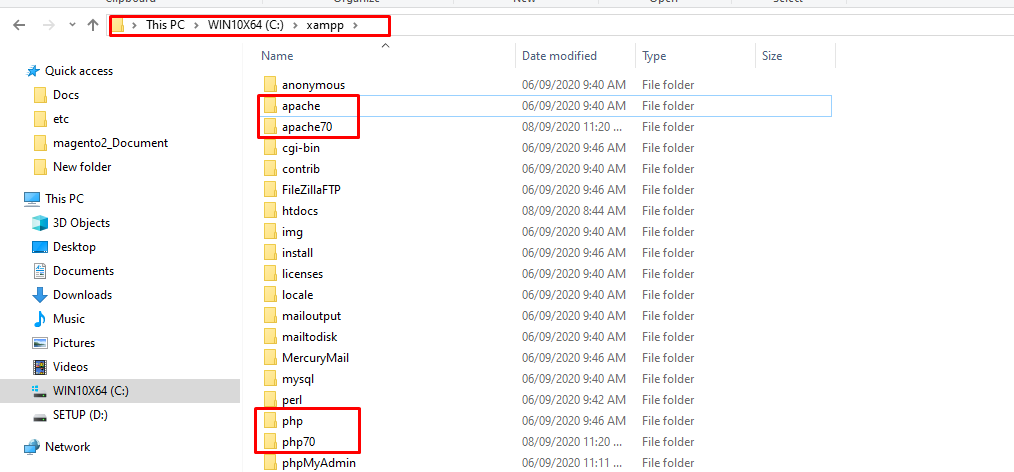


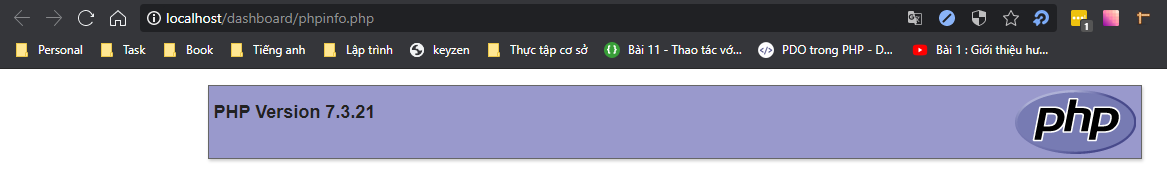
1. Lưu file php.ini lại và chạy lại apache để xem kết quả



1. Bây giờ nếu muốn chuyển đổi sang các version khác mà ta đã cài đặt ta chỉ cần đổi tên file php và apache là đã chuyển đổi được sang các version khác nhau.

VD: php73 -> php, apache73->apache



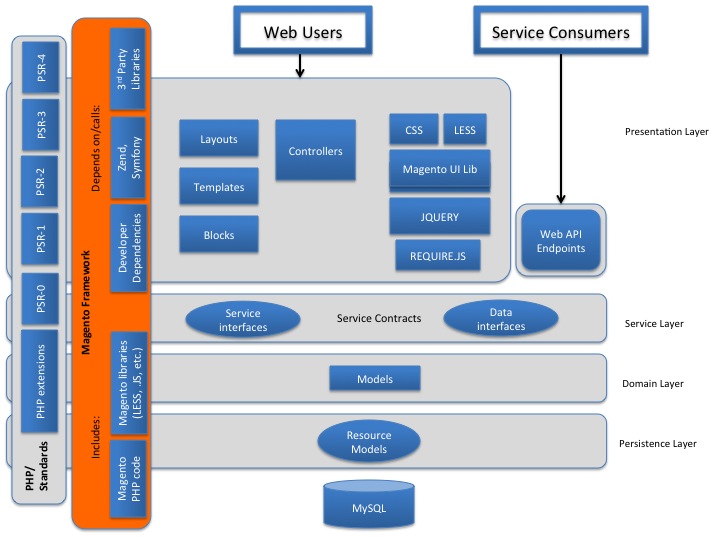


#### Nhiều bản magento cùng một php version trong một xampp

## Kiến trúc magento 2

### Kiến trúc chung

#### Mô hình các tầng



* **Presentation Layer** là lớp trên cùng của tầng 4 lớp theo mô hình kiến trúc của magento 2.
  + Trong lớp này chứa các ***layouts, blocks, templates và controllers*** xử lý các lệnh đến và đi từ giao diện người dùng.
  + Về cơ bản, lớp trình bày thể hiện sự tùy chỉnh của các tệp HTML, CSS , JavaScript, Magento UI, PHTML và các block file
  + Lớp này tiếp xúc trực tiếp với người dùng web, người quản trị,…
* **Service Layer** nằm bên dưới lớp Presentation Layer và bên trên lớp Domain Layer, chứa các hợp đồng dịch vụ, xác định cách thực hiện sẽ hoạt động.
  + Cung cấp một API ổn định cho các Module khác để gọi vào.
  + Lớp này cung cấp các interface để cho các lớp phía trên giao tiếp với các lớp phía dưới mà không cần phải làm việc trực tiếp.
  + Trong lớp này chứa các ***Respositories, Data Models, Module API***
  + Những khách hàng sử dụng gồm có: Controllers, Web service bằng cách gọi SOAP, REST API, các Modules khác thông qua các file di.xml
* **Domain Layer** nằm ở giữa lớp Persistence layer và lớp Service Layer, nó chứa các module xử lý logic nghiệp vụ.
  + Lớp này chứa các ***Models*** và ***Modules*** do lập trình viên tạo ra
  + Lớp này có thể truy cập vào theo 3 cách: Theo hợp đồng, Gọi trực tiếp hoặc tự cắm vào các modules khác
  + Để tự cắm vào các modules khác ta sử dụng ***event, plugin*** hoặc files ***di.xml***
* **Persistence layer** Đây là lớp cuối cùng trong tầng 4 lớp của magento 2. Nó làm việc trực tiếp với database thông qua việc thực thi các yêu cầu ***CRUD*** và ***EAV*** từ các lớp trên

#### Route

* Định tuyến URL trong magento 2 có dạng sau:

<store-url>/<store-code>/<front-name>/<controller-name>/<action-name>

Trong đó:

<store-url>: Là đường dẫn mặc định URL. VD: abc.com,…

<store-code>: Front-end: Rỗng,

Back-end: admin: được lưu trong app/etc/env.php

<front-name>: frontname của route được khai báo trong routes.xml

<controller-name>: Tên của thư mục controller

<action-name>: Tên của file action

* + Đường dẫn tới thư mục routes.xml trong module: etc/<area>/routes.xml

|  |  |
| --- | --- |
|  | <config xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchema  Location="urn:magento:framework:App/etc/routes.xsd">  <router id="%routerId%">  <route id="%routeId%" frontName="%frontName%">  <module name="%moduleName%"/>  </route>  </router>  </config> |

* + Cấu trúc của file route.xml:

Trong đó:

%routerId%: Chỉ định tên của bộ định tuyến trong Magento. Xem các bảng tham chiếu trong [phần lớp Bộ định tuyến](https://devdocs.magento.com/guides/v2.2/extension-dev-guide/routing.html#router-class). Thường dùng là standard.

%routeId%: Là tên đầu tiên của file XML xử lý layout liên quan của nó (Vd: routeId\_controller\_action.xml)

%frontName%: Là tên được hiển thị ngoài thanh URL

%moduleName%: Là tên module được áp dụng

#### Cấu trúc thư mục

* App: Chứa các phần mở rộng, chứa custom code
  + Code: Chứa các module custom
  + Design: Chứa các theme custom
  + Etc: Chứa phần cấu hình
    - di.xml: chứa các chỉ dẫn cho Object Manager làm việc. Mỗi module có các file di.xml riêng, chúng sẽ được gộp lại thành 1 khi magento chạy.
    - Env.php: chứa đường dẫn tới link admin, thông tin kết nối csdl,…
* Vendor: Chứa các source code mặc định, do composer quản lý. Ta không được sửa code ở đây.
  + Magento
    - Framework: chứa các lớp cơ bản, là nền tảng để xây dựng ứng dụng.
    - Module ngôn ngữ
    - Modules chức năng
    - Module theme
* Bin: Chứa các câu lệnh CLI
  + Magento: nơi để chạy các lệnh CLI
* Generated: Nơi chứa tất cả các mã đã tạo của magento. Nó sẽ tự động tạo các class factory nếu như lớp đó có hàm contructor
  + Code: sinh code cho Factory, Proxy, Plugin (hay Interceptor)
* Lib: Chứa các thư viện. Đây chủ yếu là mã hệ thống hỗ trợ magento chạy
  + Internal: chứa các thư viện phía server-side và fonts
  + Web: chứa các thư viện client-side (JavaScript) libraries, jquery
* Pub: Thư mục này chứa các tệp tĩnh được tạo ra từ magento theme của coder
* Var: Chứa các lớp tạm tời đang tạo khi chạy magento, bộ nhớ cache, các phiên, sao lưu database, các lỗi,…

### Kiến trúc của một module

* **Block**: Chứa các lớp chịu trách nhiệm chuẩn bị dữ liệu từ các lớp dưới để hiển thị thông qua template lên trang web. Nội dung của hầu hết mọi trang Magento 2 được hiển thị thông qua các lớp block.
* **Controller**: Chứa các lớp điều khiển php
* **etc**: Chứa các tệp cấu hình, đây là thư mục lõi của module chứa file **module.xml** 1 file bắt buộc.
* **Model**: Chứa các lớp mô hình, các mô hình này chịu trách nhiệm làm việc với database, xuất/nhập thông tin từ database và xử lý nó.
* **Setup**: Chứa các lớp giúp cấu trúc database module và thiết lập dữ liệu, các hành động này được thực hiện trong quá trình cài đặt hoặc nâng cấp.
* **view**: Chứa các file template, file CSS và JS, file layout, mẫu email.
* Api: Chứa những lớp PHP được hiển thị với API.
* Console: Chứa các lớp chịu trách nhiệm thực hiện các lênh CLI (php bin/magento) trên Magento 2.
* Cron: Chứa các lớp sử dụng để khởi chạy các tác vụ cron.
* CustomerData: Chứa các lớp phần trong Magento 2, được sử dụng để lưu trữ thông tin của người dùng, các thông tin được tải qua ajax làm tăng tốc độ tải trang.
* Helper: Chứa chức năng tổng hợp.
* i18n: Chứa các tệp CSV để đa ngôn ngữ hóa module.
* Observer: Chứa các lớp quan sát phản ứng với các sự kiện khác nhau (vd: tạo đơn hàng) để thực hiện một số hoạt động bổ sung.
* Plugin: Chứa các plugin cần thiết.
* UI: Chứa các lớp thành phần UI được sử dụng để tiếp nhận, xử lý và hiển thị dữ liệu trong bảng quản trị của trang web.

### Kiến trúc của một theme

## Widgets

## Các lệnh CLI cơ bản

## Các phần của trang quản trị

## Các phần của trang font-end

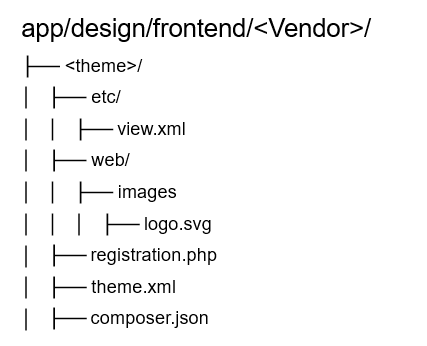
## Các bài tập làm quen magento 2

# Front-end

## Các thành phần của theme

## Các thành phần chính gồm Block, Layouts, Teamplates

* Luma là chủ đề con của Blank nên không có sự khác biệt khi kế thừa 2 chủ đề này.
* Cấu trúc một theme cơ bản:



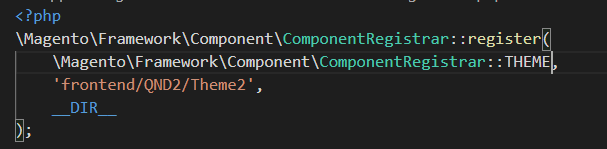
## Themes

### Tạo một theme

**Bước 1**: Trong app/design/frontend tạo thư mục Vendor: QND2 và tạo thư mục theme: Theme2

**Bước 2**: Trong folder Theme2 tạo file tragistration.php và composer.json để đăng ký theme, Tạo file theme.xml để xác định cấu hình cơ bản của theme

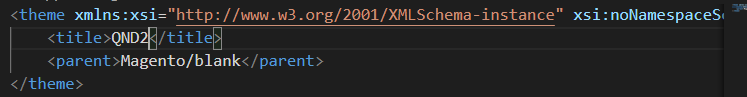
* tragistration.php



* composer.json



* Theme.xml



* Xóa cache và truy cập Content > Theme trong trang quản trị ta sẽ thấy theme mà chúng ta vừa tạo

### Tính kế thừa trong magento 2

### Xóa một theme

(Task: Làm xong phần **2.2, 1.2, 1.3,** 1.4)

## UI Component

### Tạo một UI Conponent

### UI Component gird

### Các UI Component form

#### Text field

#### Email field

#### Date, time field

#### Dropdown field

#### Radio field

#### Multiple select

#### File field

## Layout

## Templates

## HTML

## CSS

### Nhúng CSS vào magento 2

### Responsive Trong Magento 2

### Biên dịch LESS bằng grunt

#### Cài đặt grunt trong magento 2

#### Viết LESS trong magento 2

## JS

### Nhúng JS vào Magento 2

### RequireJS

### MixinJS

### KnockoutJS

## Hướng dẫn sửa một trang front-end

### Cách sắp xếp các khối block

### Chèn thêm một khối block mới

### Xóa các khối block

### Sửa một khối block có sẵn

## Hướng dẫn tạo một trang front-end mới

## Hướng dẫn chèn một plugin bên ngoài

## Bài tập sửa trang product

(2 tuần)

# Back-end

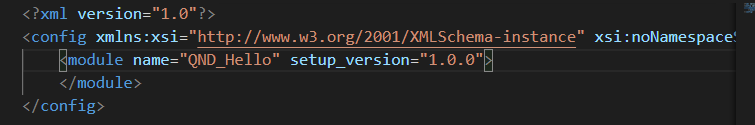
## Các thành phần của một module

## Hướng dẫn tạo một module

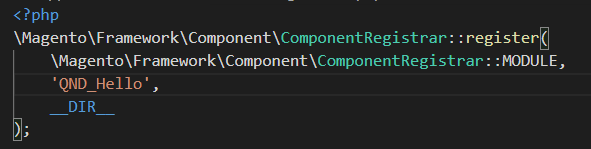
### Ngoài front-end

**Bước 1**: Tạo thư mục app/code/QND/Hello trong đó QND là Vendor name, Hello là Module name

**Bước 2:** Trong file Hello tạo file etc/module.xml để xác định cấu hình module mới.



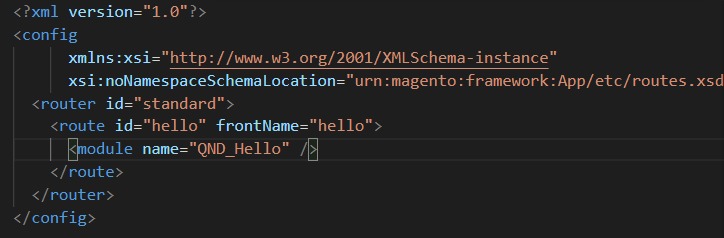
**Bước 3:** Thêm tập tin registration.php để khai báo module mới với magento 2



**Bước 4:** Kích hoạt module

* Ta chạy lệnh: php bin/magento module:status để biết tình trạng các module và thấy rằng module QND\_Hello đang bị vô hiệu hóa
* Để kích hoạt ta chạy lệnh: php bin/magento module:enable Module\_Name hoặc vào app/etc/config.php sửa ‘Module\_Name’ => 1 là xong.
* Chạy lệnh: php bin/magento setup:upgrade để nâng cấp database lên.
* Chạy thêm lệnh: php bin/magento setup:static-content:deploy –f để cập nhật các khối static của magento

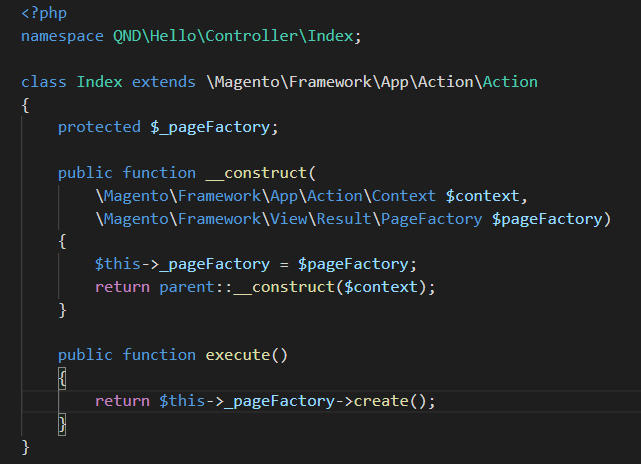
**Bước 5**: Tạo etc/frontend/routes.xml để định tuyến đường dẫn module



* Sau khi tạo route đường dẫn sẽ có dạng: localhost/magento2/hello/.

**Bước 6:** Tạo thư mục Controller

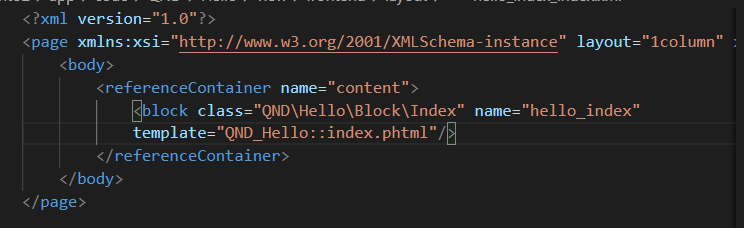
* Tạo file Controller/Index/Index.php để gọi tập tin layout.xml



* Phải khai báo PageFactory và tạo nó trong phương thức execute để hiển thị view
* Phương thức execute là phương thức dùng để thực thi câu lệnh
* Khi người dùng truy cập vào đường dẫn magento sẽ dẫn tới file action của thư mục controller để xử lý, thu nhận thông tin và chuyển hướng người dùng.

**Bước 7:** Tạo View

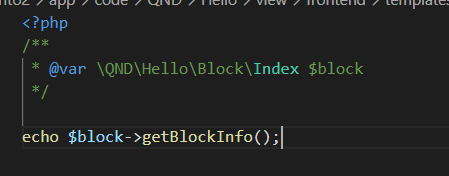
* Tạo Block, Layouts và Templates cho trang website
* Tạo layouts với đường dẫn view/frontend/layout/hello\_index\_index.xml với định dạng routeId\_nameController\_activeController.xml



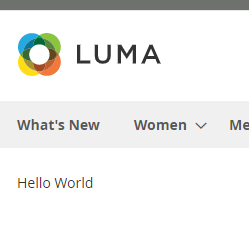
* Sau khi thư mục controller chuyển hướng người dùng đến thư mục layout thì các file layout sẽ định dạng bố cục của trang web trong magento 2
* Tạo Block với đường dẫn Block\Index.php. Block chứa các function logic có nhiệm vụ lấy dữ liệu từ Model lên cho các tầng trên



* Tạo Templates với đường dẫn view/frontend/templates/index.phtml để tạo nội dung cho trang website



* Trong file này ta có thể gọi biến $block cho đối tượng block
* Truy cập vào đường dẫn: localhost/magento2/hello/index/index để xem kết quả.



### Trong trang quản trị

Để tạo một module trong trang quản trị của magento 2 thì ta cũng làm tương tự như tạo module ngoài front-end.

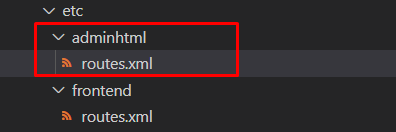
**Bước 1**: Tạo thư mục app/code/QND/Hello

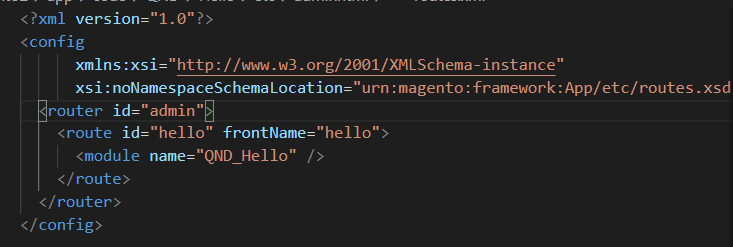
**Bước 2:** Tạo file etc/module.xml

**Bước 3:** Thêm tập tin registration.php để khai báo module mới với magento 2

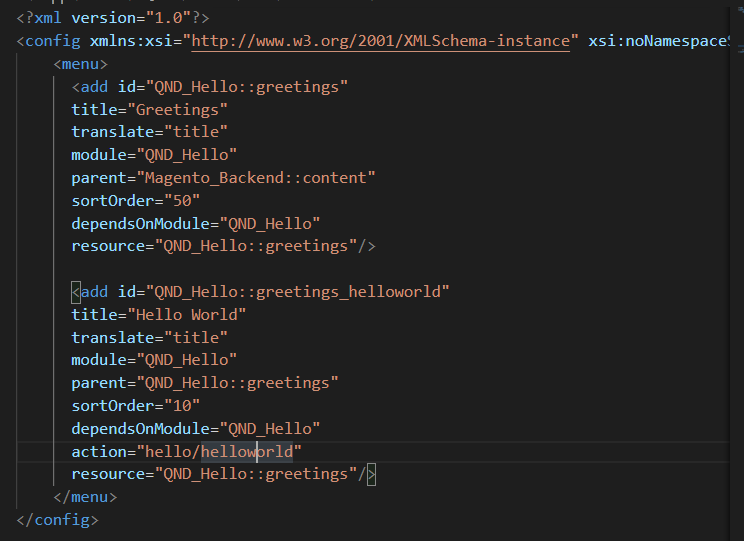
**Bước 4:** Kích hoạt module

**Bước 5**: Tạo etc/adminhtml/routes.xml để định tuyến đường dẫn module trong trang quản trị



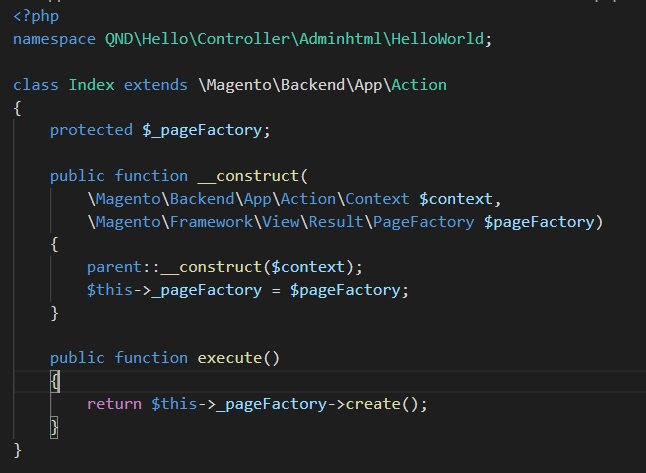


* Tạo etc/adminhtml/menu.xml



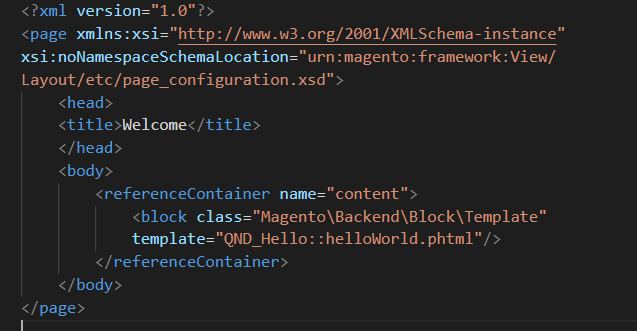
**Bước 6:** Tạo controller

* Tạo fileController/Adminhtml/HelloWorld/Index.php để điều hướng trang web

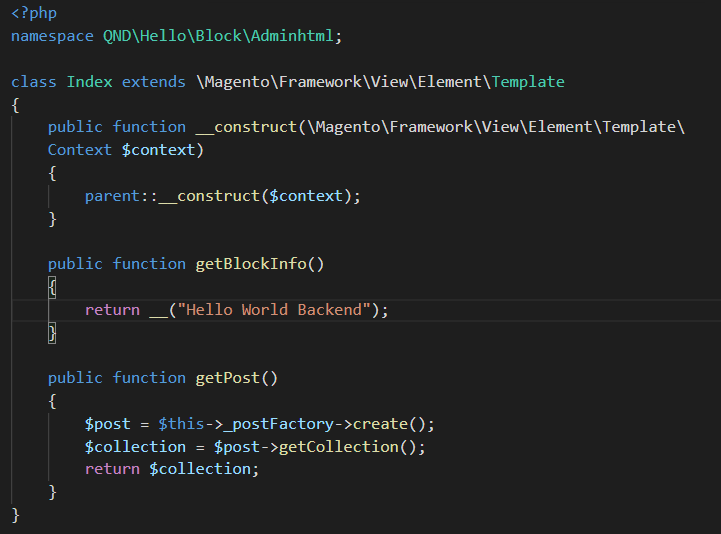


**Bước 7:** Tạo View

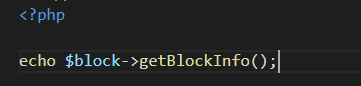
* Tạo mới 2 thư mục : view/adminhtml/layout và view/adminhtml/templates
* Trong thư mục view/adminhtml/layout tạo file hello\_helloworld\_index.xml



* Thêm tập tin Block/Adminhtml/Index.php



* Trong thư mục view/adminhtml/templates tạo file helloWorld.phtml



### Xóa module

* Để xóa module ta xóa file app/code/<Vendor>/<Module>
* Xóa các bảng của module trong database
* Xóa dữ liệu của module được lưu trong bảng core\_config\_data
* Xóa <Vendor>/<Module> nằm trong app/etc/config.php
* Xóa <Vendor>/<Module> trong bảng setup\_module
* Sau đó chạy lệnh:

php bin/magento module:uninstall -r <Vendor>\_<Module>

## Module Product

## Module Customer

## PHP Developer Guide…

(2 tuần)

# Kết nối database

## CRUD

## DI

## Object Manager

## Factory

## Model

## Resource Model

## Collection

## Responsitory

## EAV

# API