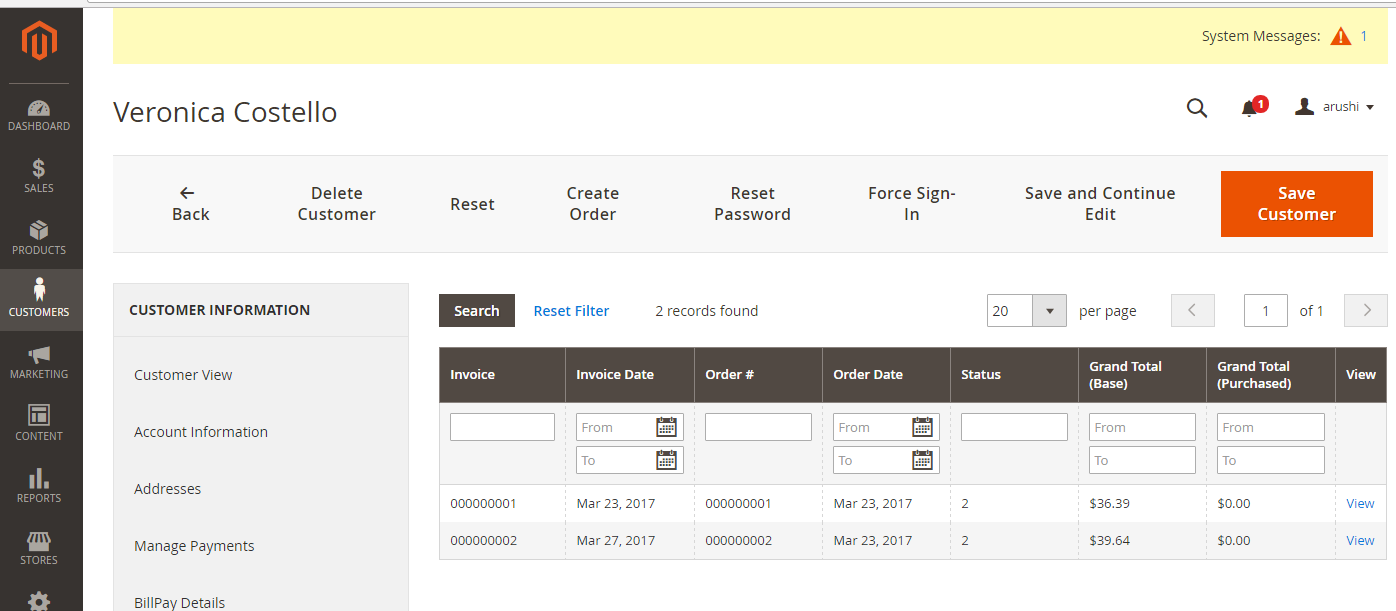
# ADD A GRID TO THE CUSTOM TAB IN CUSTOMER EDIT PAGE

Tạo thêm các tab vào customer edit



# Create Events

* Dưới etc/thư mục là cấu hình có thể được sử dụng trong cả quản trị và giao diện người dùng.
* Trong etc/frontendthư mục sẽ được sử dụng cho khu vực lối vào.
* Trong etc/adminhtmlthư mục sẽ được sử dụng cho khu vực quản trị.

Điều tương tự với tệp cấu hình sự kiện. Bạn có thể tạo tệp cấu hình sự kiện cho từng khu vực như sau:

* Khu vực quản trị: app/code/Banner/Manager/etc/adminhtml/events.xml
* Khu vực giao diện người dùng:  app/code/Banner/Manager/etc/frontend/events.xml
* Khu vực toàn cầu: app/code/Mageplaza/HelloWorld/etc/events.xml

## Gửi sự kiện

Bây giờ chúng tôi muốn gửi một danh sách sự kiện magento 2 cho phép các mô-đun khác có thể thay đổi từ được hiển thị. Chúng ta sẽ thay đổi controller như sau:

Controller/Banner/Index.php

<?php  
  
**namespace** Banner\Manager\Controller\Banner;  
  
**class** Index **extends** \Magento\Framework\App\Action\Action  
{  
  
 **public function** execute()  
 {  
 $textDisplay = **new** \Magento\Framework\DataObject(**array**('text' => 'HLW'));  
 $this->\_eventManager->dispatch('banner\_manager\_banner\_test', ['mp\_text' => $textDisplay]);  
 **echo** $textDisplay->getText();  
 **exit**;  
 }  
}

Phương thức dispatch sẽ nhận được 2 đối số: một tên sự kiện duy nhất và một dữ liệu mảng.

## Tạo sự kiện

/etc/frontend/event.xml

<?xml version="1.0"**?>**<config xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="urn:magento:framework:Event/etc/events.xsd">  
 <event name="banner\_manager\_banner\_test">  
 <observer name="bm\_display\_text" instance="Banner\Manager\Observer\ChangeDisplayText" />  
 </event>  
</config>

Trong tệp này, trong phần tử cấu hình, chúng tôi xác định phần tử sự kiện có tên là tên sự kiện được gửi ở trên. Lớp sẽ thực thi sự kiện này sẽ được định nghĩa trong phần tử Observer bởi thuộc tính instance. Tên của người quan sát được sử dụng để xác định điều này với các nhà quan sát khác của sự kiện này.

Với events.xmltệp này , Magento sẽ thực hiện lớp Mageplaza\HelloWorld\Observer\ChangeDisplayTextbất cứ khi nào phương thức gửi của sự kiện này được gọi trên khu vực lối vào. Xin lưu ý rằng, chúng tôi đặt events.xmltrong khu vực lối vào, vì vậy nếu bạn gửi sự kiện đó trong khu vực quản trị (như bộ điều khiển quản trị), nó sẽ không chạy.

## Tạo Observer

/Observer/ChangeDisplayText.php

<?php  
**namespace** Banner\Manager\Observer;  
  
**class** ChangeDisplayText **implements** \Magento\Framework\Event\ObserverInterface  
{  
 **public function** execute(\Magento\Framework\Event\Observer $observer)  
 {  
 $displayText = $observer->getData('mp\_text');  
 **echo** $displayText->getText() . " - Event </br>";  
 $displayText->setText('Execute event successfully.');  
  
 **return** $this;

}  
}

## Tạo routers

<?xml version="1.0"**?>**<!--  
/\*\*  
 \* Copyright © Magento, Inc. All rights reserved.  
 \* See COPYING.txt for license details.  
 \*/  
-->  
<config xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="urn:magento:framework:App/etc/routes.xsd">  
 <router id="standard">  
 <route id="manager" frontName="manager">  
 <module name="Banner\_Manager" before="Magento\_Backend" />  
 </route>  
 </router>  
</config>

Xóa cache và chay trên trình duyệt

# Chạy và tạo cron job

Vui lòng làm theo hướng dẫn để bắt đầu chương trình công việc cron như mong muốn của bạn trong khi lịch biểu cron được bạn đặt.

* Tạo một lớp trong thư mục “Cron”
* Cài đặt thủ công lịch biểu cron bằng cách sử dụng PHP: bin / magento cron: run
* Tìm một bản ghi trong var / log / system.log sau khi cronjob đã chạy.
* Đăng nhập vào Magento 2 Bảng điều khiển quản trị, làm như đường dẫn: Cửa hàng> Cấu hình> Nâng cao> Hệ thống, sau đó thay đổi cài đặt lịch biểu cho mỗi nhóm cron.
* Cuối cùng, chạy cron từ dòng lệnh:

magento cron:run [--group="<cron group name>"]

Bây giờ, chúng ta sẽ thêm một cron tùy chỉnh trong mô-đun HelloWorld.

## Tạo crontab.xml

/etc/crontab.xml

<?xml version="1.0" **?>**<config xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="urn:magento:module:Magento\_Cron:etc/crontab.xsd">  
 <group id="default">  
 <job instance="Vendo\Module\Cron\Test" method="execute" name="mageplaza\_helloworld\_cron">  
 <schedule>\* \* \* \* \*</schedule>  
 </job>  
 </group>  
</config>

* group id là tên nhóm cron của bạn. Bạn chỉ có thể chạy cron cho từng nhóm một.
* job instance là lớp được khởi tạo (classpath).
* job method là phương thức job instance để gọi.
* job name là ID duy nhất cho công việc cron này.
* Schedule là lịch biểu ở định dạng cron. Biểu đồ sau đây cho thấy những gì nó bao gồm:

\* \* \* \* \* command to be executed

| | | | |

| | | | +----- Day of week (0 - 7) (Sunday=0 or 7) tuần

| | | +------- Month (1 - 12) tháng

| | +--------- Day of month (1 - 31) ngày

| +----------- Hour (0 - 23) giờ

+------------- Minute (0 - 59) phút

Trong crontab.xml, chúng tôi đã xác định job instance là  \Cron\Test

## Tạo Test.php

tập tin:  /Cron/Test.php

<?php

namespace Mageplaza\HelloWorld\Cron;

class Test

{

public function execute()

{

$writer = new \Zend\Log\Writer\Stream(BP . '/var/log/cron.log');

$logger = new \Zend\Log\Logger();

$logger->addWriter($writer);

$logger->info(\_\_METHOD\_\_);

return $this;

}

}

Xóa cache và chạy php bin/magento cron:run --group="default"

Để kiểm tra xem Cron có hoạt động đúng hay không, hãy đến var/log/cron.log cửa hàng của bạn và bạn sẽ thấy văn bản  \Cron\Test::execute trong đó

# Tạo thêm dòng lệnh Command line in **console CLI**

Tạo tệp di.xml

Trong di.xml tệp, bạn có thể sử dụng một loại có tên  Magento\Framework\Console\CommandList để xác định tùy chọn lệnh.

Tập tin: app/code/Mageplaza/HelloWorld/etc/di.xml

<?*xml version*="1.0"**?>**<config xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="urn:magento:framework:ObjectManager/etc/config.xsd">  
 <type name="Magento\Framework\Console\CommandList">  
 <arguments>  
 <argument name="commands" xsi:type="array">  
 <item name="exampleSayHello" xsi:type="object">Vendo\Module\Console\Sayhello</item>  
 </argument>  
 </arguments>  
 </type>  
</config>

Cấu hình này sẽ khai báo một lớp lệnh Sayhello. Lớp này sẽ định nghĩa tên lệnh và execute()phương thức cho lệnh này.

Tập tin: app/code/Mageplaza/HelloWorld/Console/Sayhello.php

<?*php***namespace** Mageplaza\HelloWorld\Console;  
  
**use** Symfony\Component\Console\Command\Command;  
**use** Symfony\Component\Console\Input\InputInterface;  
**use** Symfony\Component\Console\Output\OutputInterface;  
  
**class** Sayhello **extends** Command  
{  
 **protected function** configure()  
 {  
 $this->setName('example:sayhello');  
 $this->setDescription('Demo command line');  
  
 **parent**::*configure*();  
 }  
 **protected function** execute(InputInterface $input, OutputInterface $output)  
 {  
 $output->writeln("Hello World");  
 }  
}

Trong hàm này, chúng ta sẽ định nghĩa 2 phương thức:

* configure()phương pháp được sử dụng để thiết lập tên, mô tả, đối số dòng lệnh của Magento 2 thêm [dòng lệnh](https://www.mageplaza.com/magento-2-command-line-interface-cli.html)
* execute() sẽ chạy khi chúng ta gọi dòng lệnh này thông qua giao diện điều khiển.

Sau khi khai báo lớp này, hãy xóa Magento cache và gõ lệnh sau:

php bin/magento --list

bin/magento example:sayhello: chạy lệnh để thực hiện

thêm các đối số cho lệnh:

<?*php***namespace** Mageplaza\HelloWorld\Console;  
  
**use** Symfony\Component\Console\Command\Command;  
**use** Symfony\Component\Console\Input\InputInterface;  
**use** Symfony\Component\Console\Output\OutputInterface;  
**use** Symfony\Component\Console\Input\InputOption;  
  
**class** Sayhello **extends** Command  
{  
  
 **const** *NAME* = 'name';  
  
 **protected function** configure()  
 {  
  
 $options = [  
 **new** InputOption(  
 **self**::*NAME*,  
 **null**,  
 InputOption::*VALUE\_REQUIRED*,  
 'Name'  
 )  
 ];  
  
 $this->setName('example:sayhello')  
 ->setDescription('Demo command line')  
 ->setDefinition($options);  
  
 **parent**::*configure*();  
 }  
  
 **protected function** execute(InputInterface $input, OutputInterface $output)  
 {  
 **if** ($name = $input->getOption(**self**::*NAME*)) {  
  
 $output->writeln("Hello " . $name);  
  
  
 } **else** {  
  
 $output->writeln("Hello World");  
  
 }  
  
 **return** $this;  
  
 }  
}

Chúng tôi đã xác định name đối số cho dòng lệnh trong configure() hàm và làm cho nó execute() hoạt động. Hãy xóa bộ nhớ cache và chạy php bin/magento example:sayhello --name="Join" từ dòng lệnh để kiểm tra kết quả. Nó sẽ hiển thị “Hello Join”

Sử dụng giao diện dòng lệnh

Trong hướng dẫn này, chúng ta sẽ nói về **Command Line Inteface (CLI) trong Magento 2** . Như bạn đã biết, từ Magento 2, họ thêm nhiều lệnh vào bin/magento. Điều này có thể khó khăn để có được cách tiếp cận này, nhưng hãy để tôi giải thích chi tiết hơn trong [hướng dẫn](https://www.mageplaza.com/kb/magento-2-tutorial/)này .

Khi bạn chạy lệnh trong terminal:

php bin/magento

hoặc là

bin/magento

Bạn sẽ nhận được danh sách dòng lệnh Magento 2 có sẵn, danh sách này bao gồm [dòng lệnh tùy chỉnh](https://www.mageplaza.com/magento-2-module-development/magento-2-how-to-add-command-line-console-cli.html)

Usage:

command [options] [arguments]

Options:

--help (-h) Display this help message

--quiet (-q) Do not output any message

--verbose (-v|vv|vvv) Increase the verbosity of messages: 1 for normal output, 2 for more verbose output and 3 for debug

--version (-V) Display this application version

--ansi Force ANSI output

--no-ansi Disable ANSI output

--no-interaction (-n) Do not ask any interactive question

Available commands:

help Displays help for a command

list Lists commands

admin

admin:user:create Creates an administrator

admin:user:unlock Unlock Admin Account

cache

cache:clean Cleans cache type(s)

cache:disable Disables cache type(s)

cache:enable Enables cache type(s)

cache:flush Flushes cache storage used by cache type(s)

cache:status Checks cache status

catalog

catalog:images:resize Creates resized product images

catalog:product:attributes:cleanup Removes unused product attributes.

cron

cron:run Runs jobs by schedule

customer

customer:hash:upgrade Upgrade customer's hash according to the latest algorithm

deploy

deploy:mode:set Set application mode.

deploy:mode:show Displays current application mode.

dev

dev:source-theme:deploy Collects and publishes source files for theme.

dev:tests:run Runs tests

dev:urn-catalog:generate Generates the catalog of URNs to \*.xsd mappings for the IDE to highlight xml.

dev:xml:convert Converts XML file using XSL style sheets

i18n

i18n:collect-phrases Discovers phrases in the codebase

i18n:pack Saves language package

i18n:uninstall Uninstalls language packages

indexer

indexer:info Shows allowed Indexers

indexer:reindex Reindexes Data

indexer:reset Resets indexer status to invalid

indexer:set-mode Sets index mode type

indexer:show-mode Shows Index Mode

indexer:status Shows status of Indexer

info

info:adminuri Displays the Magento Admin URI

info:backups:list Prints list of available backup files

info:currency:list Displays the list of available currencies

info:dependencies:show-framework Shows number of dependencies on Magento framework

info:dependencies:show-modules Shows number of dependencies between modules

info:dependencies:show-modules-circular Shows number of circular dependencies between modules

info:language:list Displays the list of available language locales

info:timezone:list Displays the list of available timezones

maintenance

maintenance:allow-ips Sets maintenance mode exempt IPs

maintenance:disable Disables maintenance mode

maintenance:enable Enables maintenance mode

maintenance:status Displays maintenance mode status

module

module:disable Disables specified modules

module:enable Enables specified modules

module:status Displays status of modules

module:uninstall Uninstalls modules installed by composer

sampledata

sampledata:deploy Deploy sample data modules

sampledata:remove Remove all sample data packages from composer.json

sampledata:reset Reset all sample data modules for re-installation

setup

setup:backup Takes backup of Magento Application code base, media and database

setup:config:set Creates or modifies the deployment configuration

setup:cron:run Runs cron job scheduled for setup application

setup:db-data:upgrade Installs and upgrades data in the DB

setup:db-schema:upgrade Installs and upgrades the DB schema

setup:db:status Checks if DB schema or data requires upgrade

setup:di:compile Generates DI configuration and all missing classes that can be auto-generated

setup:install Installs the Magento application

setup:performance:generate-fixtures Generates fixtures

setup:rollback Rolls back Magento Application codebase, media and database

setup:static-content:deploy Deploys static view files

setup:store-config:set Installs the store configuration

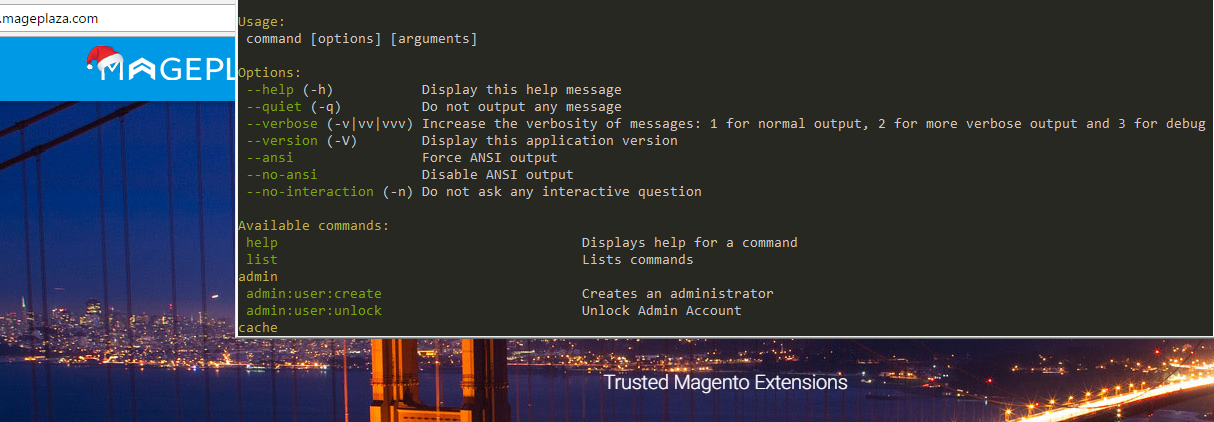
setup:uninstall Uninstalls the Magento application

setup:upgrade Upgrades the Magento application, DB data, and schema

theme

theme:uninstall Uninstalls theme

Hoặc bạn có thể thấy tương tự như ảnh chụp màn hình này



## **Danh mục dòng lệnh**

Chúng tôi có thể chia danh sách thành các danh mục sau:

* [Bộ nhớ cache](https://www.mageplaza.com/kb/how-flush-enable-disable-cache.html)
* [Người lập chỉ mục](https://www.mageplaza.com/devdocs/magento-2-reindex-indexer.html)
* [công việc lương thấp](https://www.mageplaza.com/devdocs/magento-2-create-cron-job/)
* Trình biên dịch mã
* [Đặt chế độ Magento](https://www.mageplaza.com/devdocs/magento-2-developer-mode-product-mode.html)
* i18n dịch
* [Chạy thử nghiệm đơn vị](https://www.mageplaza.com/devdocs/magento-2-unit-test/#step-2-run-unit-test)
* Bảo trì
* quản lý mô-đun
* Thiết lập: Cài đặt, Nâng cấp, Gỡ cài đặt, Sao lưu, Rollback
* Chủ đề
* Triển khai

Tiếp theo, chúng ta sẽ lấy một vài ví dụ về mỗi thể loại.

## **Dòng lệnh bộ nhớ cache**

##### **Cách kiểm tra trạng thái của (các) loại bộ nhớ cache**

php bin/magento cache:status

Kết quả:

config: 1

layout: 1

block\_html: 1

collections: 1

db\_ddl: 1

eav: 1

full\_page: 1

translate: 1

config\_integration: 1

config\_integration\_api: 1

config\_webservice: 1

##### **Cách xóa (các) loại bộ nhớ cache**

php bin/magento cache:clean

Kết quả:

$ php bin/magento cache:clean

Cleaned cache types:

config

layout

block\_html

collections

reflection

db\_ddl

eav

customer\_notification

full\_page

config\_integration

config\_integration\_api

translate

config\_webservice

##### **Cách loại bỏ (các) loại bộ nhớ cache**

php bin/magento cache:flush

##### **Cách kiểm tra bật, tắt (các) loại bộ nhớ cache**

php bin/magento cache:enable

php bin/magento cache:disable

lệnh nâng cao:

magento cache:enable [type] ... [type]

magento cache:disable [type] ... [type]

Ví dụ về vô hiệu hóa bộ nhớ cache trang đầy đủ

magento cache:disable full\_page

## **Dòng lệnh chỉ mục**

Kiểm tra trạng thái chỉ mục bằng lệnh sau

php bin/magento indexer:info

Chúng tôi sẽ nhận được kết quả:

design\_config\_grid Design Config Grid

customer\_grid Customer Grid

catalog\_category\_product Category Products

catalog\_product\_category Product Categories

catalog\_product\_price Product Price

catalog\_product\_attribute Product EAV

catalogsearch\_fulltext Catalog Search

cataloginventory\_stock Stock

catalogrule\_rule Catalog Rule Product

catalogrule\_product Catalog Product Rule

##### **Cách dữ liệu Reindexe**

php bin/magento indexer:reindex

##### **Cách đặt lại trạng thái chỉ mục thành không hợp lệ**

php bin/magento indexer:reset

##### **Cách hiển thị chế độ chỉ mục**

php bin/magento indexer:show-mode

Kết quả:

$ php bin/magento indexer:show-mode

Design Config Grid: Update on Save

Customer Grid: Update on Save

Category Products: Update on Save

Product Categories: Update on Save

Product Price: Update on Save

Product EAV: Update on Save

Catalog Search: Update on Save

Stock: Update on Save

Catalog Rule Product: Update on Save

Catalog Product Rule: Update on Save

##### **Cách đặt loại chế độ chỉ mục**

php bin/magento indexer:set-mode {realtime|schedule} [indexer]

Ví dụ:

magento indexer:set-mode schedule catalog\_category\_product catalog\_product\_category

## **Triển khai dòng lệnh**

Cú pháp lệnh:

php bin/magento setup:static-content:deploy <lang> ... <lang> [--dry-run]

#### **Cách chạy nội dung tĩnh triển khai**

php bin/magento setup:static-content:deploy

Nó sẽ hiển thị như sau:

$ php bin/magento setup:static-content:deploy

Requested languages: en\_US

=== frontend -> Magento/blank -> en\_US ===

........................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................

Theo mặc định, nó sẽ chạy triển khai nội dung tĩnh của ngôn ngữ mặc định: en\_EN

Nếu bạn muốn triển khai nội dung tĩnh của ngôn ngữ khác như: pt\_BR

Chạy

php bin/magento setup:static-content:deploy pt\_BR

Sau đó nhận kết quả:

Requested languages: pt\_BR

=== frontend -> Magento/luma -> pt\_BR ===

... progress indicator ...

Successful: 1613 files; errors: 0

=== frontend -> Magento/blank -> pt\_BR ===

... progress indicator ...

Successful: 1620 files; errors: 0

=== adminhtml -> Magento/backend -> pt\_BR ===

... progress indicator ...

Successful: 1626 files; errors: 0

=== Minify templates ===

... progress indicator ...

Successful: 858 files modified

---

## **Cách nhận đường dẫn quản trị qua dòng lệnh**

Bạn có biết? Bạn có thể nhận đường dẫn quản trị mà không cần truy cập vào etc/env.phptệp.

Chạy

php bin/magento info:adminuri

Kết quả:

$ php bin/magento info:adminuri

Admin URI: /admin

Vì vậy, đường dẫn là /admin, bây giờ điều hướng trình duyệt của bạn để [domain.com/admin](http://domain.com/admin) để truy cập vào Magento 2 phụ trợ.[miền . com / admin](http://domain.com/admin) để truy cập vào phần phụ trợ Magento 2.

## **Cách bật chế độ bảo trì qua dòng lệnh**

#### **Bật chế độ bảo trì**

php bin/magento maintenance:enable

Kết quả:

$ php bin/magento maintenance:enable

Enabled maintenance mode

#### **Tắt chế độ bảo trì**

php bin/magento maintenance:disable

#### **Cho phép các IP truy cập vào cửa hàng**

Đặt IP được miễn bảo trì chế độ bảo trì

php bin/magento maintenance:allow-ips

Ví dụ:

php bin/magento maintenance:allow-ips 168.168.168.168

Set exempt IP-addresses: 168.168.168.168

## **Cách đặt chế độ Nhà phát triển Magento hoặc Chế độ sản phẩm**

Hiển thị chế độ Magento 2 hiện tại

php bin/magento deploy:mode:show

Kết quả:

$ php bin/magento deploy:mode:show

Current application mode: default. (Note: Environment variables may override this value.)

Hiện tại, chúng tôi đang ở chế độ mặc định, hãy chuyển sang chế độ Nhà phát triển

php bin/magento deploy:mode:set developer

Kết quả:

$ php bin/magento deploy:mode:set developer

Enabled developer mode.

# **Thêm thuộc tính khách hàng**

## **Bước 1: Tạo tệp thiết lập InstallData.php**

Đầu tiên, chúng ta sẽ tạo InstallData.php file:

Tập tin: app/code/Mageplaza/HelloWorld/Setup/InstallData.php

<?php

namespace Mageplaza\HelloWorld\Setup;

use Magento\Eav\Setup\EavSetup;

use Magento\Eav\Setup\EavSetupFactory;

use Magento\Framework\Setup\InstallDataInterface;

use Magento\Framework\Setup\ModuleContextInterface;

use Magento\Framework\Setup\ModuleDataSetupInterface;

class InstallData implements InstallDataInterface

{

private $eavSetupFactory;

public function \_\_construct(EavSetupFactory $eavSetupFactory)

{

$this->eavSetupFactory = $eavSetupFactory;

}

}

Trong lớp này, chúng ta định nghĩa mô hình thiết lập EAV sẽ được sử dụng để tương tác với [thuộc tính Magento 2](https://www.mageplaza.com/magento-2-module-development/magento-2-add-customer-attribute-programmatically.html) .

## **Bước 2: Xác định phương thức install ()**

Sau đó, chúng ta phải định nghĩa install()phương thức và tạo mô hình thiết lập eav:

public function install(ModuleDataSetupInterface $setup, ModuleContextInterface $context)

{

$eavSetup = $this->eavSetupFactory->create(['setup' => $setup]);

}

Next, we will use eavSetup object to add attribute:

public function install(ModuleDataSetupInterface $setup, ModuleContextInterface $context)

{

$eavSetup = $this->eavSetupFactory->create(['setup' => $setup]);

$eavSetup->addAttribute(

\Magento\Customer\Model\Customer::ENTITY,

'sample\_attribute',

[

'type' => 'varchar',

'label' => 'Sample Attribute',

'input' => 'text',

'required' => false,

'visible' => true,

'user\_defined' => true,

'position' => 999,

'system' => 0,

]

);

}

## **Bước 3: Tạo thuộc tính tùy chỉnh**

Cuối cùng, chúng ta cần thiết lập các biểu mẫu trong đó các thuộc tính sẽ được sử dụng. Trong bước này, chúng ta cần định nghĩa đối tượng eavConfig cho phép chúng ta gọi lại thuộc tính và thiết lập dữ liệu cho nó. Và mã đầy đủ để tạo thuộc tính khách hàng là:

Tập tin: app/code/Mageplaza/HelloWorld/Setup/InstallData.php

<?php

namespace Mageplaza\HelloWorld\Setup;

use Magento\Eav\Setup\EavSetup;

use Magento\Eav\Setup\EavSetupFactory;

use Magento\Framework\Setup\InstallDataInterface;

use Magento\Framework\Setup\ModuleContextInterface;

use Magento\Framework\Setup\ModuleDataSetupInterface;

use Magento\Eav\Model\Config;

use Magento\Customer\Model\Customer;

class InstallData implements InstallDataInterface

{

private $eavSetupFactory;

public function \_\_construct(EavSetupFactory $eavSetupFactory, Config $eavConfig)

{

$this->eavSetupFactory = $eavSetupFactory;

$this->eavConfig = $eavConfig;

}

public function install(ModuleDataSetupInterface $setup, ModuleContextInterface $context)

{

$eavSetup = $this->eavSetupFactory->create(['setup' => $setup]);

$eavSetup->addAttribute(

\Magento\Customer\Model\Customer::ENTITY,

'sample\_attribute',

[

'type' => 'varchar',

'label' => 'Sample Attribute',

'input' => 'text',

'required' => false,

'visible' => true,

'user\_defined' => true,

'position' => 999,

'system' => 0,

]

);

$sampleAttribute = $this->eavConfig->getAttribute(Customer::ENTITY, 'sample\_attribute');

// more used\_in\_forms ['adminhtml\_checkout','adminhtml\_customer','adminhtml\_customer\_address','customer\_account\_edit','customer\_address\_edit','customer\_register\_address']

$sampleAttribute->setData(

'used\_in\_forms',

['adminhtml\_customer']

);

$sampleAttribute->save();

}

}

Bây giờ, hãy chạy dòng comman để cài đặt module: php bin/magento setup:upgradevà sau php bin/magento setup:static-content:deploy đó kiểm tra kết quả. Nó sẽ hiển thị như thế này:

