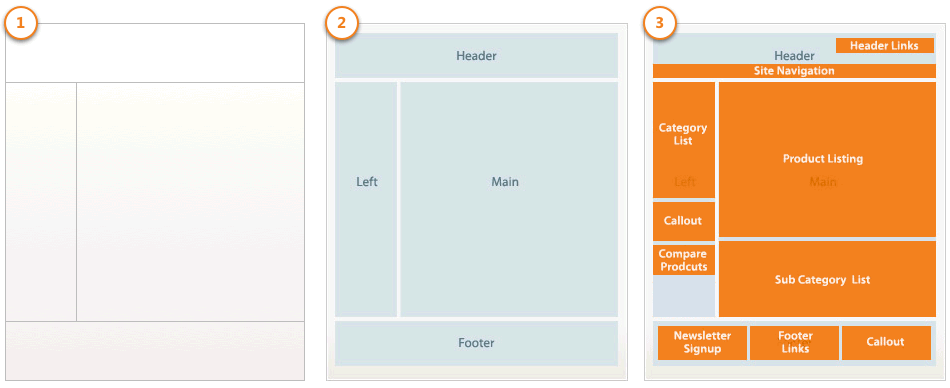
# Layout

## Giới thiệu

Các thành phần cơ bản của thiết kế trang là layout, container và các block. Một layout đại diện cho cấu trúc của một trang page (1). container đại diện cho trình giữ chỗ trong cấu trúc trang web đó (2). Và các block đại diện cho các điều khiển giao diện người dùng hoặc các thành phần trong phần giữ chỗ của container (3). Các thuật ngữ này được minh họa và được định nghĩa bên dưới.



(1) layout  cung cấp cấu trúc cho các trang web bằng cách sử dụng file XML xác định tất cả các vùng chứa và các khối soạn thảo trang. Chi tiết về file XML bố cục được mô tả sau trong phần này.

(2) Các container  gán cấu trúc nội dung cho một trang bằng cách sử dụng các thẻ chứa trong một file XML bố trí. Vùng chứa không có nội dung bổ sung ngoại trừ nội dung của các phần tử được bao gồm. Ví dụ về vùng chứa bao gồm tiêu đề, cột bên trái, cột chính và chân trang.

(3) Các block hiển thị các phần tử giao diện người dùng trên một trang bằng cách sử dụng các thẻ khối trong file XML bố trí. Chặn sử dụng các template để tạo HTML để chèn vào khối cấu trúc chính của nó. Ví dụ về các khối bao gồm danh sách danh mục, giỏ hàng nhỏ, thẻ sản phẩm và danh sách sản phẩm.

## <block>

là 1 khối (vùng ) đặc biệt của page

Các khối sử dụng các template để tạo HTML. Ví dụ về các khối bao gồm danh sách danh mục, giỏ hàng nhỏ, thẻ sản phẩm và danh sách sản phẩm.

Note : Các classthuộc tính không còn cần thiết trong các phiên bản 2.2.1trở lên vì nó sẽ mặc định Magento\Framework\View\Element\Template. **Trong các phiên bản thấp hơn 2.2.1, classthuộc tính vẫn được yêu cầu**

* Dung để tạo ra HTML

|  |  |  |  |
| --- | --- | --- | --- |
| **Thuộc tính** | **Sự miêu tả** | **Giá trị** | **Cần thiết?** |
| class | Tên của lớp thực hiện hiển thị của một khối cụ thể. Một đối tượng của lớp này chịu trách nhiệm cho việc thực hiện kết quả đầu ra khối. | tên lớp | Không |
| name | Tên có thể được sử dụng để xử lý khối mà thuộc tính này được gán. Tên phải là duy nhất trên mỗi trang được tạo. Nếu không được chỉ định, tên tự động sẽ được gán theo định dạngANONYMOUS\_*n* | 0-9, AZ, az, dấu gạch dưới (\_), dấu chấm (.), Dấu gạch ngang (-). Nên bắt đầu bằng một chữ cái. Trường hợp nhạy cảm. | Không |
| before | Được sử dụng để định vị khối  trước một phần tử trong cùng một phụ huynh. Tên phần tử hoặc tên bí danh được chỉ định trong giá trị. Sử dụng dấu gạch ngang (-) để định vị khối trước tất cả các phần tử khác của cấp độ lồng của nó. Xem [trước và sau các thuộc tính](https://devdocs.magento.com/guides/v2.2/frontend-dev-guide/layouts/xml-instructions.html#fedg_xml-instrux_before-after) để biết chi tiết. | Giá trị có thể có: tên phần tử hoặc dấu gạch ngang (-) | Không |
| after | Được sử dụng để định vị khối sau khi một phần tử trong cùng một cha mẹ. Tên phần tử hoặc tên bí danh được chỉ định trong giá trị. Sử dụng dấu gạch ngang (-) để định vị khối sau khi tất cả các phần tử khác của cấp độ lồng của nó. Xem [trước và sau các thuộc tính](https://devdocs.magento.com/guides/v2.2/frontend-dev-guide/layouts/xml-instructions.html#fedg_xml-instrux_before-after)để biết chi tiết. | Giá trị có thể có: tên phần tử hoặc dấu gạch ngang (-) | Không |
| template | Một template đại diện cho chức năng của khối mà thuộc tính này được gán. | tên file template | Không |
| as | Tên bí danh đóng vai trò là số nhận dạng trong phạm vi của phần tử gốc. | 0-9, AZ, az, dấu gạch dưới (\_), dấu chấm (.), Dấu gạch ngang (-). Trường hợp nhạy cảm. | Không |
| cacheable | Xác định xem một phần tử khối có thể lưu vào bộ nhớ đệm hay không. Điều này có thể được sử dụng cho các mục đích phát triển và để tạo ra các yếu tố cần thiết của trang động. | true hoặc là false | Không |

Để truyền tham số, sử dụng thẻ [<argument></argument>](https://devdocs.magento.com/guides/v2.2/frontend-dev-guide/layouts/xml-instructions.html#argument).

## <container>

Cấu trúc không có nội dung chứa các phần tử bố cục khác như block và container.

container hiển thị phần tử con trong quá trình tạo đầu ra của chế độ xem. Nó có thể trống hoặc nó có thể chứa một bộ <container>và <block>các phần tử tùy ý

|  |  |  |  |
| --- | --- | --- | --- |
| **Thuộc tính** | **Sự miêu tả** | **Giá trị** | **Cần thiết?** |
| name | Tên có thể được sử dụng để giải quyết vùng chứa trong đó thuộc tính này được chỉ định. Tên phải là duy nhất trên mỗi trang được tạo. | AZ, az, 0-9, dấu gạch dưới (\_), dấu chấm (.), Dấu gạch ngang (-). Nên bắt đầu bằng một chữ cái. Trường hợp nhạy cảm. | Vâng |
| label | Một tên tùy ý để hiển thị trong trình duyệt web. | bất kì | Không |
| before | Được sử dụng để định vị vùng chứa trước một phần tử trong cùng một cha mẹ. Tên phần tử hoặc tên bí danh được chỉ định trong giá trị. Sử dụng dấu gạch ngang (-) để định vị khối trước tất cả các phần tử khác của cấp độ lồng của nó. Xem [trước và sau các thuộc tính](https://devdocs.magento.com/guides/v2.2/frontend-dev-guide/layouts/xml-instructions.html#fedg_xml-instrux_before-after) để biết chi tiết. | Giá trị có thể có: tên phần tử hoặc dấu gạch ngang (-). | Không |
| after | Được sử dụng để định vị vùng chứa sau một phần tử trong cùng một cha mẹ. Tên phần tử hoặc tên bí danh được chỉ định trong giá trị. Sử dụng dấu gạch ngang (-) để định vị khối sau khi tất cả các phần tử khác của cấp độ lồng của nó. Xem [trước và sau các thuộc tính](https://devdocs.magento.com/guides/v2.2/frontend-dev-guide/layouts/xml-instructions.html#fedg_xml-instrux_before-after)để biết chi tiết. | Giá trị có thể có: tên phần tử hoặc dấu gạch ngang (-). | Không |
| as | Tên bí danh đóng vai trò là số nhận dạng trong phạm vi của phần tử gốc. | 0-9, AZ, az, dấu gạch dưới (\_), dấu chấm (.), Dấu gạch ngang (-). Trường hợp nhạy cảm. | Không |
| output | Xác định xem có xuất phần tử gốc hay không. Nếu được chỉ định, phần tử sẽ được thêm vào danh sách đầu ra. (Nếu không được chỉ định, phần tử gốc có trách nhiệm hiển thị con của nó.) | Bất kỳ giá trị nào ngoại trừ lỗi thời toHtml. Giá trị được đề xuất là 1. | Không |
| htmlTag | Tham số đầu ra. Nếu được chỉ định, đầu ra được đưa vào thẻ HTML được chỉ định. | Bất kỳ thẻ HTML 5 hợp lệ nào. | Không |
| htmlId | Tham số đầu ra. Nếu được chỉ định, giá trị được thêm vào phần tử trình bao bọc. Nếu không có phần tử bao bọc, thuộc tính này không có hiệu lực. | Bất kỳ giá trị HTML 5 hợp lệ nào <id>. | Không |
| htmlClass | Tham số đầu ra. Nếu được chỉ định, giá trị được thêm vào phần tử trình bao bọc. Nếu không có phần tử bao bọc, thuộc tính này không có hiệu lực. | Bất kỳ giá trị HTML 5 hợp lệ nào <class>. | Không |

Ví dụ

<container name="div.sidebar.additional" htmlTag="div" htmlClass="sidebar sidebar-additional" after="div.sidebar.main">

<container name="sidebar.additional" as="sidebar\_additional" label="Sidebar Additional"/>

</container>

Điều này sẽ thêm cột mới vào bố cục trang. ### trước và sau thuộc tính {# fedg\_xml-instrux\_before-after}

Để giúp bạn định vị các phần tử theo thứ tự cụ thể phù hợp với thiết kế, SEO, khả năng sử dụng hoặc các yêu cầu khác, phần mềm Magento cung cấp các thuộc tính beforevà afterbố cục.

Các thuộc tính tùy chọn này có thể được sử dụng trong các file XML bố trí để kiểm soát thứ tự của các phần tử trong cha mẹ chung của chúng. bảng sau đây cho một mô tả chi tiết về kết quả mà bạn có thể nhận được bằng cách sử dụng beforevà afterthuộc tính. Bảng đầu tiên sử dụng một khối làm phần tử định vị.

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** |  | **Value** | **Description** |
| before |  | Dash (-) | Block hiển thị trước tất cả các phần tử khác trong nút cha của nó.. |
| before |  | [element name] | block được hiển thị trước phần tử được đặt tên. |
| before |  | empty value or [element name] is absent | Sử dụng giá trị của after. Nếu giá trị đó trống hoặc không có, thì phần tử được coi là không được định vị. |
| after |  | Dash (-) | Được hiển thị sau tất cả các phần tử cùng cấp |
| after |  | [element name] | Block được hiển thị sau phần tử được đặt tên. |
| after |  | empty value or [element name] is absent | Sử dụng giá trị của before. Nếu giá trị đó trống hoặc không có, khối đó được coi là không được định vị. |

Ví dụ

|  |  |
| --- | --- |
| **Tình hình** | **Kết quả** |
| Cả hai beforevà aftercác thuộc tính đều có mặt | after được ưu tiên. |
| Cả hai thuộc tính before và after đều không có hoặc trống | Phần tử được coi là không được định vị. Tất cả các yếu tố khác được đặt tại vị trí được chỉ định của chúng. Phần tử bị thiếu hiển thị ở vị trí ngẫu nhiên không vi phạm các yêu cầu cho các phần tử được định vị. |
| Một số phần tử có before hoặc after được đặt thành dấu gạch ngang (-) | Tất cả các phần tử hiển thị ở trên cùng (hoặc dưới cùng, trong trường hợp thuộc tính sau), nhưng thứ tự nhóm của các phần tử này không được xác định. |
| Các before hoặc after giá trị thuộc tính đề cập đến một yếu tố không nằm trong nút cha của nguyên tố này được xác định. | Phần tử hiển thị ở vị trí ngẫu nhiên không vi phạm các yêu cầu cho các phần tử được định vị chính xác. |

## <action>

Được sử dụng để thiết lập việc thực hiện một phương thức nhất định của block trong quá trình tạo block; các thẻ <action> phải được bố trí trong phạm vi của <block>.

Ví dụ

<block class="Magento\Module\Block\Class" name="block">

<action method="setText">

<argument name="text" translate="true" xsi:type="string">Text</argument>

</action>

<action method="setEnabled">

<argument name="enabled" xsi:type="boolean">true</argument>

</action>

</block>

* Method : tên phương thức của lớp mà thẻ nằm trong đó

Để chuyển các tham số, sử dụng  lệnh [<argument></argument>](https://devdocs.magento.com/guides/v2.2/frontend-dev-guide/layouts/xml-instructions.html#argument).

## <referenceBlock> và <referenceContainer>

* Cập nhật <referenceBlock>và <referenceContainer>được áp dụng cho tương ứng <block>hoặc <container>.

Ví dụ: nếu bạn thực hiện tham chiếu theo <referenceBlock name="right">, bạn đang nhắm mục tiêu cho block <block name="right">.

Để chuyển các tham số cho một khối sử dụng [<argument></argument>](https://devdocs.magento.com/guides/v2.2/frontend-dev-guide/layouts/xml-instructions.html#argument)lệnh.

* Remove : Cho phép loại bỏ hoặc hủy bỏ phần tử. Khi một container được loại bỏ, các phần tử con của nó cũng bị loại bỏ.
* display : Cho phép bạn tắt hiển thị block hoặc container cụ thể với tất cả các con của nó (cả hai được đặt trực tiếp và theo tham chiếu). Block / container và các đối tượng PHP tương ứng của con của nó vẫn được tạo và sẵn sàng cho thao tác.
* Các remove thuộc tính là tùy chọn và giá trị mặc định của nó là false.

Triển khai này cho phép bạn xóa một khối hoặc vùng chứa trong bố cục của mình bằng cách đặt giá trị thuộc tính remove thành truehoặc để hủy bỏ khối hoặc vùng chứa bằng cách đặt giá trị thành false. Thí dụ:

<referenceBlock name = "block.name" remove = "true" />

* Các displaythuộc tính là tùy chọn và giá trị mặc định của nó là sự thật.

Bạn luôn có thể ghi đè giá trị này trong bố cục của mình. Trong trường hợp khi xóa giá trị là true, thuộc tính display sẽ bị bỏ qua. Thí dụ:

<referenceContainer name = "container.name" display = "false" />

## <move>

Di chuyển phần tử đã được khai báo hoặc phần tử vùng chứa làm phần tử con của phần tử khác theo thứ tự được chỉ định

Ví dụ

<move element="name.of.an.element" destination="name.of.destination.element" as="new\_alias" after="name.of.element.after" before="name.of.element.before"/>

* <move> bị bỏ qua nếu phần tử được di chuyển không được xác định.
* Nếu as thuộc tính không được xác định, giá trị hiện tại của bí danh phần tử được sử dụng. Nếu không thể, name thay vào đó , giá trị của thuộc tính được sử dụng.
* Trong quá trình tạo bố cục, <move> lệnh được xử lý trước khi xóa (được đặt bằng remove thuộc tính). Điều này có nghĩa là nếu bất kỳ phần tử nào được chuyển đến phần tử được lên lịch để xóa, chúng cũng sẽ bị xóa.

|  |  |  |  |
| --- | --- | --- | --- |
| **Thuộc tính** | **Sự miêu tả** | **Giá trị** | **Cần thiết?** |
| thành phần | Tên của phần tử cần di chuyển. | tên phần tử | Vâng |
| destination | Tên của phần tử mẹ mục tiêu. | tên phần tử | Vâng |
| as | Tên bí danh cho phần tử ở vị trí mới. | 0-9, AZ, az, dấu gạch dưới (\_), dấu chấm (.), Dấu gạch ngang (-). Trường hợp nhạy cảm. | Không |
| after | before | Chỉ định vị trí của phần tử liên quan đến anh chị em ruột. Sử dụng dấu gạch ngang (-) để định vị khối trước hoặc sau tất cả các anh chị em khác của cấp độ lồng của nó. Nếu thuộc tính bị bỏ qua, phần tử được đặt sau tất cả các anh chị em. | tên phần tử | Không |

## <remove>

chỉ được sử dụng để loại bỏ các tài nguyên tĩnh được liên kết trong một <head>phần của trang . Để xóa các Block hoặc container, hãy sử dụng thuộc tính <remove> cho [<referenceBlock> và <referenceContainer>](https://devdocs.magento.com/guides/v2.2/frontend-dev-guide/layouts/xml-instructions.html#fedg_layout_xml-instruc_ex_ref) . Ví dụ về cách sử dụng:

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

<head>

<!-- Remove local resources -->

<remove src="css/styles-m.css" />

<remove src="my-js.js"/>

<remove src="Magento\_Catalog::js/compare.js" />

<!-- Remove external resources -->

<remove src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.4/css/bootstrap-theme.min.css"/>

<remove src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.4/js/bootstrap.min.js"/>

<remove src="http://fonts.googleapis.com/css?family=Montserrat" />

</head>

</page>

## <update>

Bao gồm một file bố cục nhất định

<update handle="{name\_of\_handle\_to\_include}"/>

Xử [lý](https://devdocs.magento.com/guides/v2.2/frontend-dev-guide/layouts/layout-overview.html#handle) được chỉ định là "được bao gồm" và được thực thi đệ quy.

## <argument>

Được sử dụng để chuyển một đối số. Phải luôn luôn được đính kèm trong [<arguments>](https://devdocs.magento.com/guides/v2.2/frontend-dev-guide/layouts/xml-instructions.html#arguments).

|  |  |  |
| --- | --- | --- |
| **Thuộc tính** | **Sự miêu tả** | **Giá trị** |
| name | Tên đối số. | duy nhất |
| xsi:type | Loại đối số. | chuỗi | boolean | đối tượng | số | null | mảng (string|boolean|object|number|null|array) |
| translate |  | đúng | sai |

Ví dụ

Để vượt qua nhiều đối số, hãy sử dụng cấu trúc sau:

<arguments>

<argument></argument>

<argument></argument>

</arguments>

Để vượt qua một đối số là một mảng sử dụng cấu trúc sau đây:

<argument>

<item></item>

<item></item>

...

</argument>

Để vượt qua một đối số là một mảng sử dụng cấu trúc sau đây:

<argument>

<item></item>

<item></item>

...

</argument>

Các giá trị đối số được đặt trong file bố cục có thể được truy cập trong các [template](https://devdocs.magento.com/guides/v2.2/frontend-dev-guide/templates/template-overview.html) bằng cách sử dụng các phương thức get{ArgumentName}()và has{ArgumentName}(). Cái sau trả về một boolean xác định xem có bất kỳ giá trị nào được đặt hay không. {ArgumentName}được lấy từ namethuộc tính theo cách sau: để lấy giá trị của <argument name="some\_string">tên phương thức getSomeString(). Ví dụ: Đặt giá trị css\_classtrong [app/code/Magento/Theme/view/frontend/layout/default.xml](https://github.com/magento/magento2/blob/2.0/app/code/Magento/Theme/view/frontend/layout/default.xml)file bố cục:

...

<arguments>

<argument name="css\_class" xsi:type="string">header links</argument>

</arguments>

...

Sử dụng giá trị của css\_classtrong [app/code/Magento/Theme/view/frontend/templates/html/title.phtml](https://github.com/magento/magento2/blob/2.0/app/code/Magento/Theme/view/frontend/templates/html/title.phtml):

...

$cssClass = $this->hasCssClass() ? ' ' . $this->getCssClass() : '';

...

### <arguments> {#arguments} <arguments>là vùng chứa bắt buộc cho <argument>. Nó không có thuộc tính riêng của nó. Thí dụ:

...

<arguments>

<argument name="css\_class" xsi:type="string">header links</argument>

</arguments>

...

## Page layout

* Khai báo bên trong <body>

<Magento\_Theme\_module\_dir>/view/frontend/page\_layout/2columns-left.xml

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

<update handle="1column"/>

<referenceContainer name="columns">

<container name="div.sidebar.main" htmlTag="div" htmlClass="sidebar sidebar-main" after="main">

<container name="sidebar.main" as="sidebar\_main" label="Sidebar Main"/>

</container>

<container name="div.sidebar.additional" htmlTag="div" htmlClass="sidebar sidebar-additional" after="div.sidebar.main">

<container name="sidebar.additional" as="sidebar\_additional" label="Sidebar Additional"/>

</container>

</referenceContainer>

</layout>

* **Khai báo layout module <module\_dir>/view/frontend/layouts.xml**
* **Khai báo layout theme <theme\_dir>/<Namespace>\_<Module>/layouts.xml**

**Ví dụ :**

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

<update handle="1column"/>

<referenceContainer name="columns">

<container name="div.sidebar.main" htmlTag="div" htmlClass="sidebar sidebar-main" after="main">

<container name="sidebar.main" as="sidebar\_main" label="Sidebar Main"/>

</container>

<container name="div.sidebar.additional" htmlTag="div" htmlClass="sidebar sidebar-additional" after="div.sidebar.main">

<container name="sidebar.additional" as="sidebar\_additional" label="Sidebar Additional"/>

</container>

</referenceContainer>

</layout>

Bảng sau đây mô tả các hướng dẫn cụ thể cho các file cấu hình trang

|  |
| --- |
|  |
| **Thành phần** | **Thuộc tính** | **Bố mẹ của** | **Sự miêu tả** |
| <page></page> | * layout = {layout} * xsi:noNamespaceSchemaLocation ="{path\_to\_schema}" | * <html> * <head> * <body> * <update> | Phần tử gốc bắt buộc. |
| <html></html> | không ai | * <attribute> | Yếu tố bắt buộc. |
| <head></head> | không ai | * <title> * <meta> * <link> * <css> * <script> |  |
| <body></body> | không ai | * <block> * <container> * <move> * <attribute> * <referenceBlock> * <referenceContainer> * <action> |  |
| <attribute> | * name = {arbitrary\_name} * value = {arbitrary\_value} |  | Được chỉ định cho <html>, được hiển thị như sau:  <html name="value'> |
| <title> | không ai | không ai | Tiêu đề trang |
| <meta> | * content * charset * http-equiv * name * scheme | không ai |  |
| <link> | * defer * ie\_condition * charset * hreflang * media * rel * rev * sizes * src * src\_type * target * type | không ai |  |
| <css> | * defer * ie\_condition * charset * hreflang * media * rel * rev * sizes * src * src\_type * target * type | không ai |  |
| <script> | * defer * ie\_condition * async * charset * src * src\_type * type | không ai |  |

## Extend layout

sửa đổi những gì muốn thay đổi, trong hệ thống Magento, chỉ cần tạo layout extends có chứa các thay đổi bạn muốn

<theme\_dir>

 |\_\_/<Namespace>\_<Module>

   |\_\_/layout

     |--<layout1>.xml

     |--<layout2>.xml

Ví dụ: để tùy chỉnh bố cục được xác định trong

**<Magento\_Catalog\_module\_dir>/view/frontend/layout/catalog\_product\_view.xml**, bạn cần phải thêm file bố cục có cùng tên trong theme tùy chỉnh của bạn, như sau:

**<theme\_dir>/Magento\_Catalog/layout/catalog\_product\_view.xml**

Để thêm file [bố cục trang](https://devdocs.magento.com/guides/v2.2/frontend-dev-guide/layouts/layout-types.html#layout-types-page) mở rộng :

1. Đặt file ở vị trí sau:

<theme\_dir>

 |\_\_/<Namespace>\_<Module>

  |\_\_/page\_layout

     |--<layout1>.xml

     |--<layout2>.xml

## Override a layout

Số lượng tùy chỉnh lớn thì cần phải thực hiện ghi đè.

### Ghi đè base layout

Vị trí

<theme\_dir>

  |\_\_/<Namespace\_Module>

    |\_\_/layout

      |\_\_/**override**

         |\_\_/base

           |--<layout1>.xml

           |--<layout2>.xml

Những file này ghi đè các bố cục sau:

* <module\_dir>/view/frontend/layout/<layout1>.xml
* <module\_dir>/view/frontend/layout/<layout2>.xml

### Ghi đè theme layout

<theme\_dir>

  |\_\_/<Namespace\_Module>

    |\_\_/layout

      |\_\_/override

         |\_\_/theme

            |\_\_/<Parent\_Vendor>

               |\_\_/<parent\_theme>

                  |--<layout1>.xml

                  |--<layout2>.xml

Những file này ghi đè các bố cục sau:

* <parent\_theme\_dir>/<Namespace>\_<Module>/layout/<layout1>.xml
* <parent\_theme\_dir>/<Namespace>\_<Module>/layout/<layout2>.xml

Note

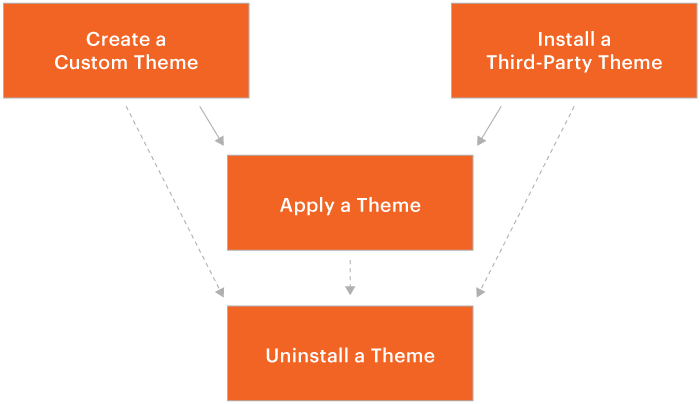
không nên thực hiện các thay đổi sau

* Thay đổi tên khối hoặc bí danh. Tên của một khối không được thay đổi, và không phải là bí danh của một khối còn lại trong cùng một phần tử cha.
* Thay đổi kế thừa xử lý. Ví dụ, bạn không nên thay đổi loại trang phụ huynh xử lý.

# Theme

## Themes overview

* Theme là 1 thành phần ứng dụng magento cung cấp giao diện trực quan cho toàn bộ ứng dụng trong magento



## Cài đặt theme mặt tiền cửa hàng của bên thứ ba

Theme này mô tả cách cài đặt theme của bên thứ ba cho cửa hàng Magento 2.

Để cài đặt một theme, bạn cần thêm mã của nó vào cơ sở mã trường hợp Magento 2 của bạn, và sau đó đăng ký nó trong cơ sở dữ liệu. Cách một theme được phân phối xác định cách thực hiện điều này:

* nếu một theme chỉ là một tập hợp các file, ví dụ như một file lưu trữ, hãy thêm theme theo cách thủ công.
* nếu một theme là một gói nhà soạn nhạc , hãy cài đặt nó bằng cách sử dụng nhà soạn nhạc.
* nếu một theme được phân phối dưới dạng phần mở rộng, hãy sử dụng **Trình hướng dẫn thiết lập web** trong Magento Admin .

Các phần sau chứa nhiều thông tin hơn về mỗi luồng cài đặt.

### Cài đặt theme theo cách thủ công

Để cài đặt theme theo cách thủ công:

1. Đảm bảo rằng cấu trúc thư mục bạn đang sao chép là <VendorName>/<theme>. Và tất cả các [tập tin theme](https://devdocs.magento.com/guides/v2.2/frontend-dev-guide/themes/theme-structure.html) nằm trong <theme>thư mục.
2. Sao chép thư mục này vào <Magento root dir>/app/design/frontendthư mục.

### Cài đặt theme làm gói nhà soạn nhạc

Để cài đặt theme làm gói trình soạn nhạc, hãy làm theo hướng dẫn trong theme [Cài đặt, quản lý và nâng cấp mô-đun](https://devdocs.magento.com/guides/v2.2/cloud/howtos/install-components.html) .

### Cài đặt theme làm tiện ích mở rộng

Nếu một theme được phân phối trên [Magento Marketplace](https://marketplace.magento.com/) , hãy cài đặt nó với **Trình hướng dẫn thiết lập web** . Để biết chi tiết, hãy xem [Cài đặt tiện ích mở rộng](http://docs.magento.com/marketplace/user_guide/quick-tour/install-extension.html) .

### Đăng ký một theme

Sau khi bạn tạo hoặc cài đặt một theme, mở Magento Admin hoặc tải lại bất kỳ trang Quản trị Magento nào. Bằng cách này, theme được đăng ký và thêm vào cơ sở dữ liệu.

Để đảm bảo theme được đăng ký, trong Magento Admin, hãy điều hướng đến **Content** > **Design** > **Themes** . Nếu một theme được đăng ký, nó sẽ được hiển thị trong danh sách các theme có sẵn.

## Gỡ cài đặt theme mặt tiền cửa hàng

Theme này mô tả cách gỡ cài đặt theme mặt tiền cửa hàng trong Magento 2.

Cách gỡ cài đặt theme được xác định bởi hai yếu tố:

* cách theme được thêm vào: thêm thủ công (đã cài đặt hoặc tạo), được cài đặt làm gói trình soạn thảo hoặc dưới dạng tiện ích mở rộng .
* cách Magento được cài đặt: [sử dụng các file nguồn từ Github](https://devdocs.magento.com/guides/v2.2/install-gde/install/cli/install-cli-sample-data-clone.html) hoặc [sử dụng Composer](https://devdocs.magento.com/guides/v2.2/install-gde/install/cli/install-cli-sample-data-composer.html) .

Các phần sau mô tả luồng để gỡ cài đặt theme trong từng trường hợp.

### Điều kiện tiên quyết

1. [Đặt ứng dụng Magento của bạn thành nhà phát triển hoặc chế độ mặc định](https://devdocs.magento.com/guides/v2.2/config-guide/cli/config-cli-subcommands-mode.html) .
2. Đảm bảo rằng theme không được áp dụng trên mặt tiền cửa hàng. Để thực hiện việc này, trong bảng điều khiển Quản trị, hãy điều hướng đến **Nội dung** > **Thiết kế** > **Cấu hình** và đảm bảo rằng theme tùy chỉnh của bạn không được áp dụng cho bất kỳ chế độ xem cửa hàng nào .
3. Hãy chắc chắn rằng theme không được định nghĩa là một phụ huynh cho bất kỳ theme đã đăng ký nào. Để thực hiện việc này, trong bảng điều khiển Quản trị, điều hướng đến **Nội dung** > **Thiết kế** > **Theme** . Đảm bảo rằng theme của bạn không được đề cập trong cột **Theme chính** . Nếu được đề cập, trước tiên bạn cần gỡ cài đặt theme con.

### Gỡ cài đặt theme được thêm theo cách thủ công

Trong trường hợp nếu theme của bạn được tạo hoặc cài đặt bằng tay, quy trình gỡ cài đặt sẽ giống nhau, bất kể cách cài đặt Magento.

Để gỡ cài đặt theme được thêm theo cách thủ công:

1. Điều hướng đến thư mục nhà cung cấp nơi theme được cài đặt. Thư mục này nên là: <Magento root dir>/app/design/frontend/<VendorName>.
2. Xóa thư mục theme.
3. Xóa bản ghi theme khỏi cơ sở dữ liệu. Nếu bạn đang sử dụng MySQL, hãy chạy lệnh sau để thực hiện việc này:

mysql -u <user> -p -e "delete from <dbname>.theme where theme\_path ='<Vendor>/<theme>' AND area ='frontend' limit 1"

Ở đâu:

* <user>: tên người dùng cơ sở dữ liệu Magento của bạn
* <dbname>: tên cơ sở dữ liệu Magento của bạn
* <Vendor>/<theme>: đường dẫn tương đối đến thư mục theme

### Gỡ cài đặt gói theme

Luồng để gỡ cài đặt theme là gói Composer là khác nhau, tùy thuộc vào cách cài đặt Magento của bạn.

### Gỡ cài đặt gói theme nếu Magento được cài đặt bằng Composer

Nếu cả theme và cá thể Magento được cài đặt bằng Composer, bạn có thể sử dụng lệnh CLI đặc biệt. Làm theo hướng dẫn từ [theme gói Gỡ cài đặt chủ](https://devdocs.magento.com/guides/v2.2/install-gde/install/cli/install-cli-theme-uninstall.html) đề.

### Gỡ cài đặt gói theme nếu Magento được cài đặt bằng cách sao chép kho lưu trữ

Để gỡ cài đặt gói Composer theme nếu phiên bản Magento của bạn được cài đặt bằng cách nhân bản kho lưu trữ git, bạn cũng có thể gỡ cài đặt nó bằng lệnh CLI. Tuy nhiên, trước tiên bạn phải xóa nó khỏi danh sách phụ thuộc.

Thực hiện các bước sau:

1. Mở <Magento root dir>/composer.jsonfile.
2. Tìm một dòng có tham chiếu đến gói theme và xóa nó. Tham chiếu sẽ giống như sau:
3. "require": {
4. ...
5. "<vendor>/<theme-name>": "<version>"
6. },
7. Để cập nhật các phụ thuộc của dự án, hãy chạy:

cập nhật soạn nhạc

1. Sử dụng magento theme:uninstalllệnh CLI như được mô tả trong [theme gói Gỡ cài đặt theme soạn thảo](https://devdocs.magento.com/guides/v2.2/install-gde/install/cli/install-cli-theme-uninstall.html) .

Bạn có thể sử dụng lệnh Composer remove để loại bỏ sự phụ thuộc, nhưng trong trường hợp đó, bạn phải xóa bản ghi theme khỏi cơ sở dữ liệu theo cách thủ công.

### Gỡ cài đặt tiện ích mở rộng theme

Nếu theme đã được cài đặt dưới dạng tiện ích mở rộng, bạn có thể gỡ cài đặt theme bằng một trong các luồng sau:

* giống như cách các gói Trình soạn thảo theme được gỡ cài đặt, hãy xem phần [Gỡ cài đặt gói theme](https://devdocs.magento.com/guides/v2.2/frontend-dev-guide/themes/theme-uninstall.html#uninstall_theme_pack) để biết chi tiết.
* bằng cách sử dụng Component Manager.

Để gỡ cài đặt tiện ích mở rộng theme bằng Trình quản lý thành phần:

1. Trong Bảng điều khiển quản trị Magento , điều hướng đến **Hệ thống** > **Thuật sĩ thiết lập web** > **Trình quản lý tiện ích mở rộng** .
2. Trong cột **Hành động** , bấm **Chọn** > **Gỡ cài đặt** trong bản ghi theme.

## Tạo theme Quản trị

### Tổng quan

Để tạo theme Quản trị tùy chỉnh , hãy thực hiện các bước sau:

1. [Tạo một thư mục theme.](https://devdocs.magento.com/guides/v2.2/frontend-dev-guide/themes/admin_theme_create.html#create_dir)
2. [Thêm tuyên bố theme.xml.](https://devdocs.magento.com/guides/v2.2/frontend-dev-guide/themes/admin_theme_create.html#declare_theme)
3. [Thêm registration.php.](https://devdocs.magento.com/guides/v2.2/frontend-dev-guide/themes/admin_theme_create.html#add_registry)
4. [Tùy chọn thêm composer.json.](https://devdocs.magento.com/guides/v2.2/frontend-dev-guide/themes/admin_theme_create.html#make_composer)
5. [Tùy ý thay đổi biểu tượng theme.](https://devdocs.magento.com/guides/v2.2/frontend-dev-guide/themes/admin_theme_create.html#logo)

Mỗi bước được mô tả thêm.

### Tạo thư mục theme

Trong app/design/adminhtmlthư mục tạo một <Vendor>/<admin\_theme>thư mục mới .

### Thêm tuyên bố theme.xml

Trong thư mục theme, thêm theme.xmlchứa ít nhất tên theme và tên theme gốc (nếu theme [kế thừa](https://devdocs.magento.com/guides/v2.2/frontend-dev-guide/themes/theme-inherit.html) từ một). Chúng tôi khuyên bạn nên kế thừa từ theme mặc định Magento Admin: Magento/backend.

Thêm hoặc sao chép từ thư mục hiện có theme.xmlvào thư mục theme của bạn app/design/adminhtml/<Vendor>/<admin\_theme>.

Định cấu hình nó bằng ví dụ sau (thay thế trình giữ chỗ bằng thông tin theme của bạn ):

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

<title>%Theme title%</title> <!-- your theme's name -->

<parent>%vendor\_dir%/%parent\_theme\_dir%</parent> <!-- the parent theme. Example: Magento/backend -->

</theme>

Nếu bạn thay đổi tiêu đề theme hoặc thông tin theme cha mẹ theme.xmlsau khi một theme đã được [đăng ký](https://devdocs.magento.com/guides/v2.2/frontend-dev-guide/themes/admin_theme_create.html#register_theme) , bạn cần mở hoặc tải lại bất kỳ trang Quản trị Magento nào để các thay đổi của bạn được lưu trong cơ sở dữ liệu.

### Thêm vào registration.xml

Trong thư mục theme của bạn, hãy tạo một registration.phpfile. Trong file này, hãy thêm mã sau đây, thay thế trình giữ chỗ bằng thông tin theme của bạn:

<?php

/\*\*

\* Copyright © 2016 Magento. All rights reserved.

\* See COPYING.txt for license details.

\*/

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

\Magento\Framework\Component\ComponentRegistrar::THEME,

'adminhtml/%vendor\_dir/your\_theme\_dir%', // Example: 'adminhtml/Magento/backend'

\_\_DIR\_\_

);

### Tùy chọn thêm composer.json

Xem gói [Làm cho theme của bạn thành một nhà soạn nhạc (tùy chọn)](https://devdocs.magento.com/guides/v2.2/frontend-dev-guide/themes/theme-create.html#fedg_create_theme_composer)

### Biểu tượng theme quản trị (tùy chọn)

Trong Magento/backendtheme mặc định lib/web/images/magento-logo.svgđược sử dụng làm biểu tượng theme. Để ghi đè nó, trong thư mục theme của bạn, tạo một web/imagesthư mục con và thêm file tùy chỉnh của bạn có tên magento-logo.svg. Nếu bạn muốn sử dụng file với tên và / hoặc định dạng khác, bạn cần khai báo bổ sung nó như được mô tả trong [Tuyên bố biểu tượng theme](https://devdocs.magento.com/guides/v2.2/frontend-dev-guide/themes/theme-create.html#logo_declare) .

### Đăng ký theme

Khi bạn mở Magento Admin (hoặc tải lại bất kỳ trang quản trị Magento nào) đã thêm các file theme vào hệ thống file, theme của bạn sẽ được đăng ký và thêm vào cơ sở dữ liệu.

## Áp dụng theme Quản trị

### Điều kiện tiên quyết

1. [Đặt](https://devdocs.magento.com/guides/v2.2/config-guide/cli/config-cli-subcommands-mode.html) ứng dụng Magento của bạn sang [chế độ](https://devdocs.magento.com/guides/v2.2/config-guide/bootstrap/magento-modes.html) nhà phát triển . Chế độ ứng dụng ảnh hưởng đến cách các file tĩnh được lưu trữ bởi Magento.
2. [Tạo theme tùy chỉnh cho bảng điều khiển Quản trị](https://devdocs.magento.com/guides/v2.2/frontend-dev-guide/themes/admin_theme_create.html) .
3. [Thêm mô-đun tùy chỉnh mới](https://devdocs.magento.com/guides/v2.2/extension-dev-guide/build/build.html) hoặc quyết định sử dụng mô-đun tùy chỉnh hiện có. Mô-đun phải tải sau mô-đun Magento\_Theme. Để đảm bảo điều này, hãy thêm mã sau vào <your\_custom\_module\_dir>/etc/module.xml(thay thế trình giữ chỗ bằng thông tin mô-đun của bạn ):
4. <module name="%YourVendor\_YourModule%" setup\_version="2.0.1"> <!-- Example: "Magento\_Backend" -->
5. <sequence>
6. <module name="Magento\_Theme"/>
7. <module name="Magento\_Enterprise"/> <!-- For Enterprise versions only -->
8. </sequence>
9. </module>

Nếu bạn chọn tạo một mô-đun chuyên dụng riêng biệt, bạn có thể sử dụng [mô-đun Magento\_SampleMinimal từ kho lưu trữ mô-đun template Magento 2](https://github.com/magento/magento2-samples/tree/master/sample-module-minimal) làm ví dụ về mô-đun tối thiểu bạn cần. Nếu bạn sẽ sao chép và sử dụng Magento\_SampleMinimal, đừng quên để vào nhà cung cấp của bạn và module đặt tên, thay vì những người sử dụng trong template, trong <your\_module\_dir>/etc/module.xml, <your\_module\_dir>/registration.php, and <your\_module\_dir>/composer.jsontập tin.

Nếu bạn quyết định sử dụng mô-đun hiện có, hãy nhớ rằng theme tuyên bố có thể bị ảnh hưởng khi mô-đun được thay đổi.

### Áp dụng theme tùy chỉnh trong Quản trị: Tổng quan

Để áp dụng theme Quản trị , hãy thực hiện các bước sau:

1. [Chỉ định theme Quản trị mới trong mô-đun của bạn di.xml](https://devdocs.magento.com/guides/v2.2/frontend-dev-guide/themes/admin_theme_apply.html#specify_di)
2. Cập nhật các thành phần bằng cách chạy [bin/magento setup:upgrade](https://devdocs.magento.com/guides/v2.2/install-gde/install/cli/install-cli-uninstall.html#instgde-install-keep)lệnh.
3. Mở Quản trị viên trong trình duyệt và xem theme mới được áp dụng.

Mỗi bước được mô tả chi tiết hơn.

### Chỉ định theme Quản trị tùy chỉnh trong di.xml

Bạn cần chỉ định theme quản trị được sử dụng trong <your\_module\_dir>/etc/adminhtml/di.xmlfile. Thêm nó, nếu tập tin chưa tồn tại trong mô-đun của bạn.

Trong <your\_module\_dir>/etc/adminhtml/di.xmladd như sau (thay thế placeholders với tên nhà cung cấp và mã theme của theme quản trị của bạn):

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

<!-- Admin theme. Start -->

<type name="Magento\Theme\Model\View\Design">

<arguments>

<argument name="themes" xsi:type="array">

<item name="adminhtml" xsi:type="string">%Your\_vendor\_dir%/%your\_theme\_code%</item> <!-- Example: "Magento/backend" -->

</argument>

</arguments>

</type>

<!-- Admin theme. End -->

</config>

### Cập nhật các thành phần để thực sự áp dụng theme Quản trị

Để các thay đổi của bạn có hiệu lực, bạn cần cập nhật các thành phần Magento. Đối với điều này, hãy chạy bin/magento setup:upgradelệnh trong dòng lệnh của bạn. Nếu được nhắc, cũng chạy bin/magento setup:di:compile.

Để biết chi tiết về cách thực hiện các tác vụ dòng lệnh, hãy xem các theme sau:

* [Cấu hình dòng lệnh](https://devdocs.magento.com/guides/v2.2/config-guide/cli/config-cli.html)
* [Gỡ cài đặt hoặc cài đặt lại Magento: Tùy chọn giữ các file đã tạo](https://devdocs.magento.com/guides/v2.2/install-gde/install/cli/install-cli-subcommands-db-upgr.html)

### Mở Quản trị viên trong trình duyệt

Bước cuối cùng là mở Quản trị viên trong trình duyệt và xem theme mới được áp dụng.

## Tạo theme cửa hàng mới

Theme này thảo luận cách tạo các file tạo nên một theme, cách thêm biểu tượng vào theme và cách định kích thước hình ảnh.

Theme mới bạn tạo không được áp dụng cho cửa hàng của bạn một cách tự động. Bạn cần áp dụng thủ công trong bảng điều khiển Quản trị. Quy trình này được mô tả trong phần [Áp dụng và định cấu hình theme trong](https://devdocs.magento.com/guides/v2.2/frontend-dev-guide/themes/theme-apply.html) theme [Quản trị](https://devdocs.magento.com/guides/v2.2/frontend-dev-guide/themes/theme-apply.html) .

### Điều kiện tiên quyết

1. Vì lợi ích của khả năng tương thích, nâng cấp, và bảo trì dễ dàng, không sửa đổi các theme hộp Magento. Để tùy chỉnh thiết kế cửa hàng Magento của bạn, hãy tạo theme tùy chỉnh mới .
2. [Đặt](https://devdocs.magento.com/guides/v2.2/config-guide/cli/config-cli-subcommands-mode.html) ứng dụng Magento của bạn sang [chế độ](https://devdocs.magento.com/guides/v2.2/config-guide/bootstrap/magento-modes.html) nhà phát triển . Chế độ ứng dụng ảnh hưởng đến cách các file tĩnh được lưu trữ bởi Magento. Các khuyến nghị về phát triển theme mà chúng tôi cung cấp trong chương này là nhà phát triển / chế độ mặc định cụ thể.

### Tạo theme mặt tiền cửa hàng: hướng dẫn

Các bước cao cấp cần thiết để thêm một theme mới trong hệ thống Magento như sau:

1. Tạo một thư mục cho theme dưới app/design/frontend/<your\_vendor\_name>/<your\_theme\_name>.
2. Thêm một file khai báo theme.xmlvà tạo etcthư mục tùy chọn và tạo một file có tên là view.xmlthư mục theme.
3. Thêm composer.jsonfile.
4. Thêm registration.php.
5. Tạo thư mục cho CSS, JavaScript, hình ảnh và phông chữ.
6. Định cấu hình theme của bạn trong bảng điều khiển Quản trị .

## Áp dụng và định cấu hình theme mặt tiền cửa hàng

Theme mô tả cách áp dụng theme cho cửa hàng của bạn. Đây là bước bắt buộc nếu bạn muốn một theme được sử dụng trên mặt tiền cửa hàng . Ngoài ra, nó cung cấp thông tin làm thế nào để thêm một biểu tượng theme độc lập cho cửa hàng của bạn.

### Điều kiện tiên quyết

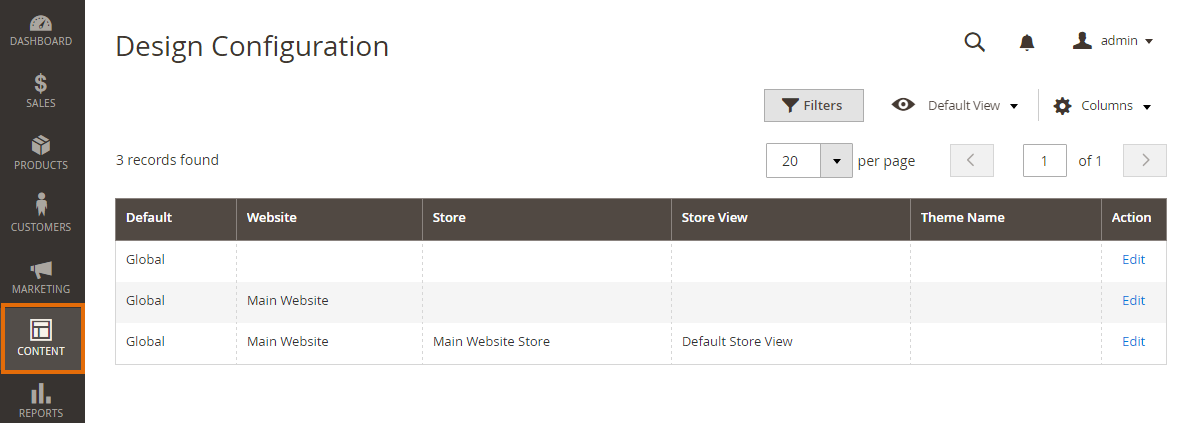
Đảm bảo rằng bạn [đã đặt](https://devdocs.magento.com/guides/v2.1/config-guide/cli/config-cli-subcommands-mode.html) ứng dụng Magento của mình thành [chế độ](https://devdocs.magento.com/guides/v2.1/config-guide/bootstrap/magento-modes.html) nhà phát triển .

### Áp dụng theme

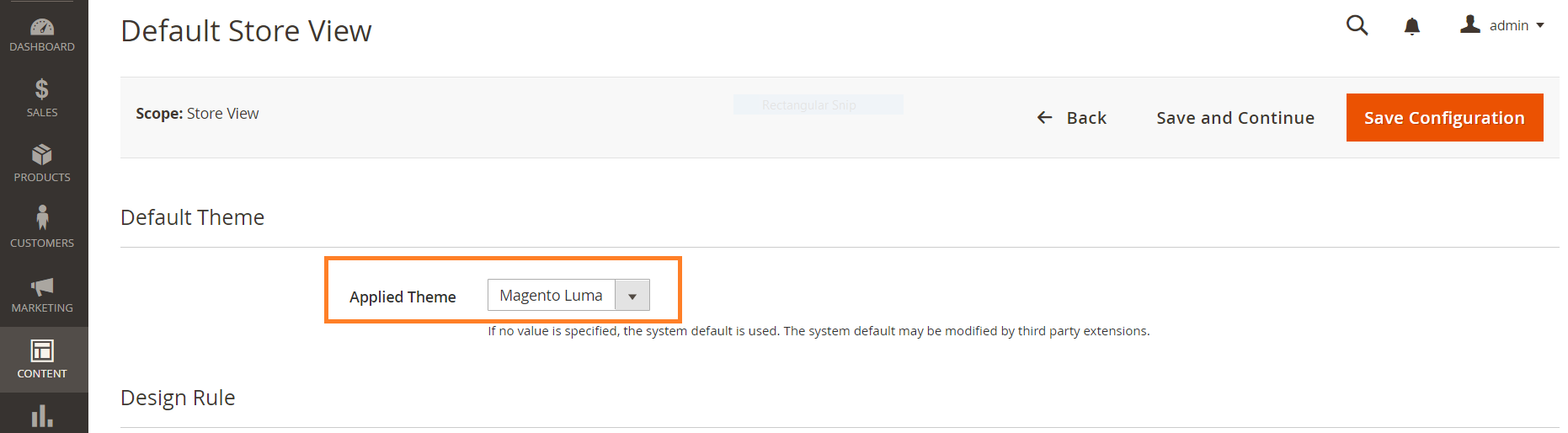
Sau khi bạn [thêm theme của mình vào hệ thống file](https://devdocs.magento.com/guides/v2.1/frontend-dev-guide/themes/theme-create.html) , bạn có thể áp dụng theme đó cho cửa hàng của mình. Bạn áp dụng một theme trong Admin .

Để áp dụng theme:

1. Trong Quản trị viên, đi tới **CONTENT** > **Design** > **Configuaration** . Một trang Cấu hình thiết kế sẽ mở ra. Nó chứa một lưới với các phạm vi cấu hình có sẵn. Ví dụ:



1. Trong bản ghi cấu hình tương ứng với chế độ xem cửa hàng của bạn, hãy nhấp vào **Chỉnh sửa** . Trang có cấu hình thiết kế cho phạm vi đã chọn sẽ mở ra. Ví dụ:



1. Trên tab **Theme mặc định** , trong menu thả xuống **Theme được áp dụng** , chọn theme mới được tạo của bạn.
2. Nhấp vào **Lưu cấu hình** .
3. Nếu bộ nhớ đệm được bật, hãy [xóa bộ nhớ cache](https://devdocs.magento.com/guides/v2.2/frontend-dev-guide/themes/theme-apply.html#theme-apply-clear) .
4. Để xem các thay đổi của bạn được áp dụng, hãy tải lại các trang mặt tiền cửa hàng.

### Thêm ngoại lệ thiết kế

Ngoại lệ thiết kế cho phép bạn chỉ định theme thay thế cho các tác nhân người dùng cụ thể, thay vì tạo chế độ xem cửa hàng riêng cho chúng. Để thêm ngoại lệ thiết kế:

1. Trong Quản trị viên, đi tới **CONTENT** > **Thiết kế** > **Cấu hình**
2. Trong bản ghi cấu hình tương ứng với chế độ xem cửa hàng của bạn, hãy nhấp vào **Chỉnh sửa** .
3. Trên tab **Quy tắc thiết kế** , bấm **Thêm quy tắc tác nhân người dùng mới** .
4. Trong hộp **Chuỗi tìm kiếm,** chỉ định tác nhân người dùng sử dụng chuỗi thông thường hoặc biểu thức chính quy (PCRE). Trong danh sách thả xuống **Tên theme,** hãy chọn theme được sử dụng cho tác nhân phù hợp.
5. Nhấp vào **Lưu cấu hình** hoặc **Lưu và tiếp tục** .
6. Nếu bộ nhớ đệm được bật, hãy [xóa bộ nhớ cache](https://devdocs.magento.com/guides/v2.2/frontend-dev-guide/themes/theme-apply.html#theme-apply-clear) .
7. Để xem các thay đổi của bạn được áp dụng, hãy tải lại các trang mặt tiền cửa hàng.

### Thêm biểu trưng độc lập cho theme

Bạn có thể muốn đặt biểu tượng cửa hàng cố định hiển thị trên mặt tiền cửa hàng bất kể theme nào được áp dụng. Để thêm biểu trưng độc lập vĩnh viễn:

1. Trong Quản trị viên, đi tới **CONTENT** > **Thiết kế** > **Cấu hình**
2. Trong bản ghi cấu hình tương ứng với chế độ xem cửa hàng của bạn, hãy nhấp vào **Chỉnh sửa** .
3. Mở rộng tab **Tiêu đề** .
4. Trong trường **Logo Image,** duyệt đến tập tin logo được lưu trong hệ thống tập tin của bạn.
5. Tải file lên.
6. Tùy chọn, chỉ định chiều rộng, chiều cao mong muốn và văn bản thay thế cho biểu tượng trong các trường tương ứng.
7. Nhấp vào **Lưu cấu hình** hoặc **Lưu và tiếp tục** .
8. Nếu bộ nhớ đệm được bật, hãy [xóa bộ nhớ cache](https://devdocs.magento.com/guides/v2.2/frontend-dev-guide/themes/theme-apply.html#theme-apply-clear) .
9. Để xem các thay đổi của bạn được áp dụng, hãy tải lại các trang mặt tiền cửa hàng.

Biểu trưng bạn thêm ở đây được lưu trữ trong /pub/media/logo/default/thư mục.

Để xóa biểu trưng vĩnh viễn, hãy chuyển đến cùng một vị trí và nhấp vào biểu tượng “Xóa hình ảnh” ở góc dưới cùng bên trái của bản xem trước biểu trưng.

### Xóa bộ nhớ cache

Nếu bộ nhớ đệm được kích hoạt trong Magento Admin, bạn phải xóa bộ nhớ cache sau khi bạn áp dụng theme, thêm một ngoại lệ thiết kế, thêm logo và thực hiện các tác vụ khác.

Một thông báo hệ thống thông báo cho bạn rằng các loại bộ nhớ cache không hợp lệ phải được làm mới.

1. Nhấp vào **Hệ thống** > **Quản lý bộ nhớ cache** .
2. Xóa các loại bộ nhớ cache không hợp lệ.

### Khắc phục sự cố (nếu các thay đổi không được áp dụng)

Nếu các thay đổi bạn định cấu hình trong Quản trị viên không được áp dụng sau khi bạn xóa bộ nhớ cache và tải lại trang, xóa tất cả các file trong thư mục pub/static/frontendvà var/view\_preprocessedrồi tải lại các trang. Bạn có thể xóa các tập tin bằng tay hoặc chạy grunt clean:<theme\_name>lệnh trong CLI. Để biết chi tiết về cách sử dụng Grunt trong Magento

## Định vị template, bố cục và kiểu

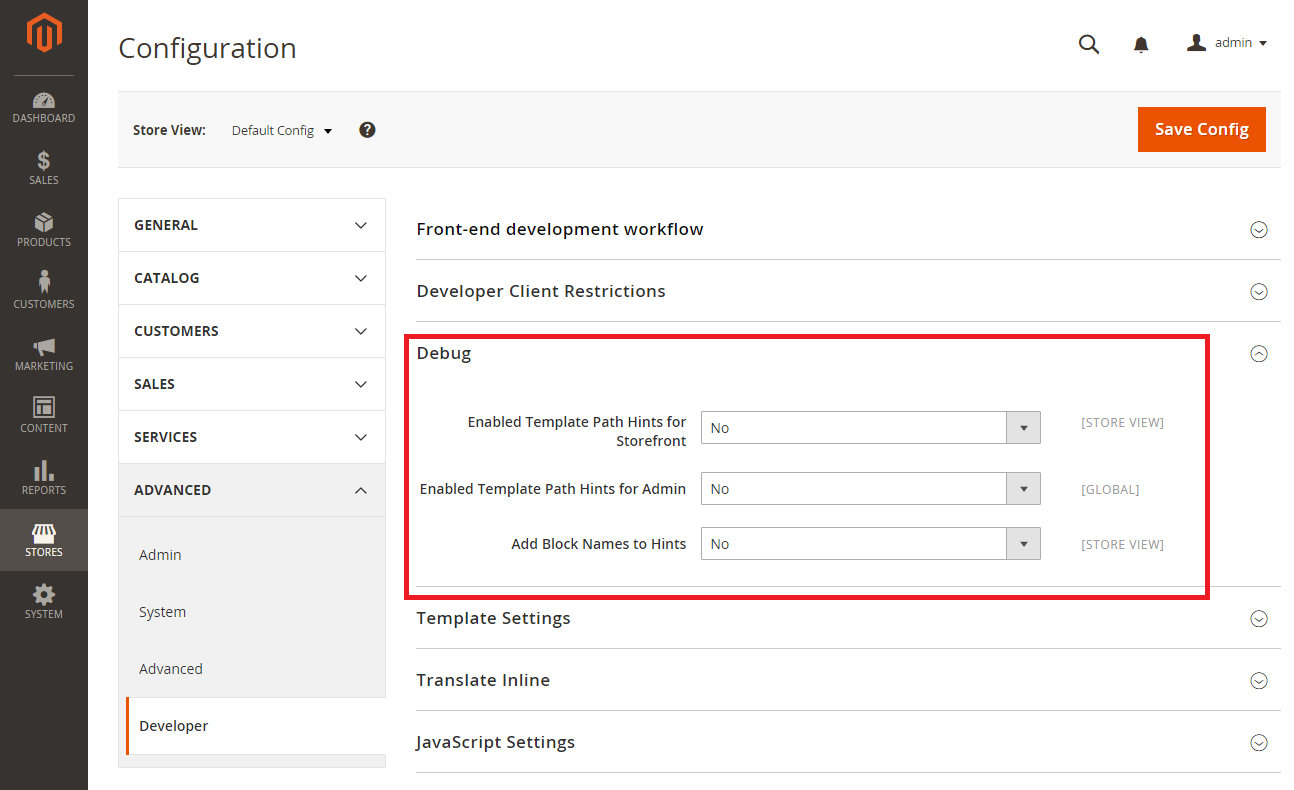
Khi bạn tạo theme Magento, bạn có thể cần tạo các file ghi đè cho các file chế độ xem theme và mô-đun mặc định . Để làm như vậy, bạn phải xác định các template, bố cục và kiểu file nào mà Magento sử dụng. Theme này mô tả cách thực hiện điều này.

### Định vị template

Để định vị khuôn template chịu trách nhiệm cho một phần cụ thể của mặt tiền cửa hàng hoặc Quản trị viên, bạn có thể sử dụng các gợi ý template được xây dựng trong Magento.

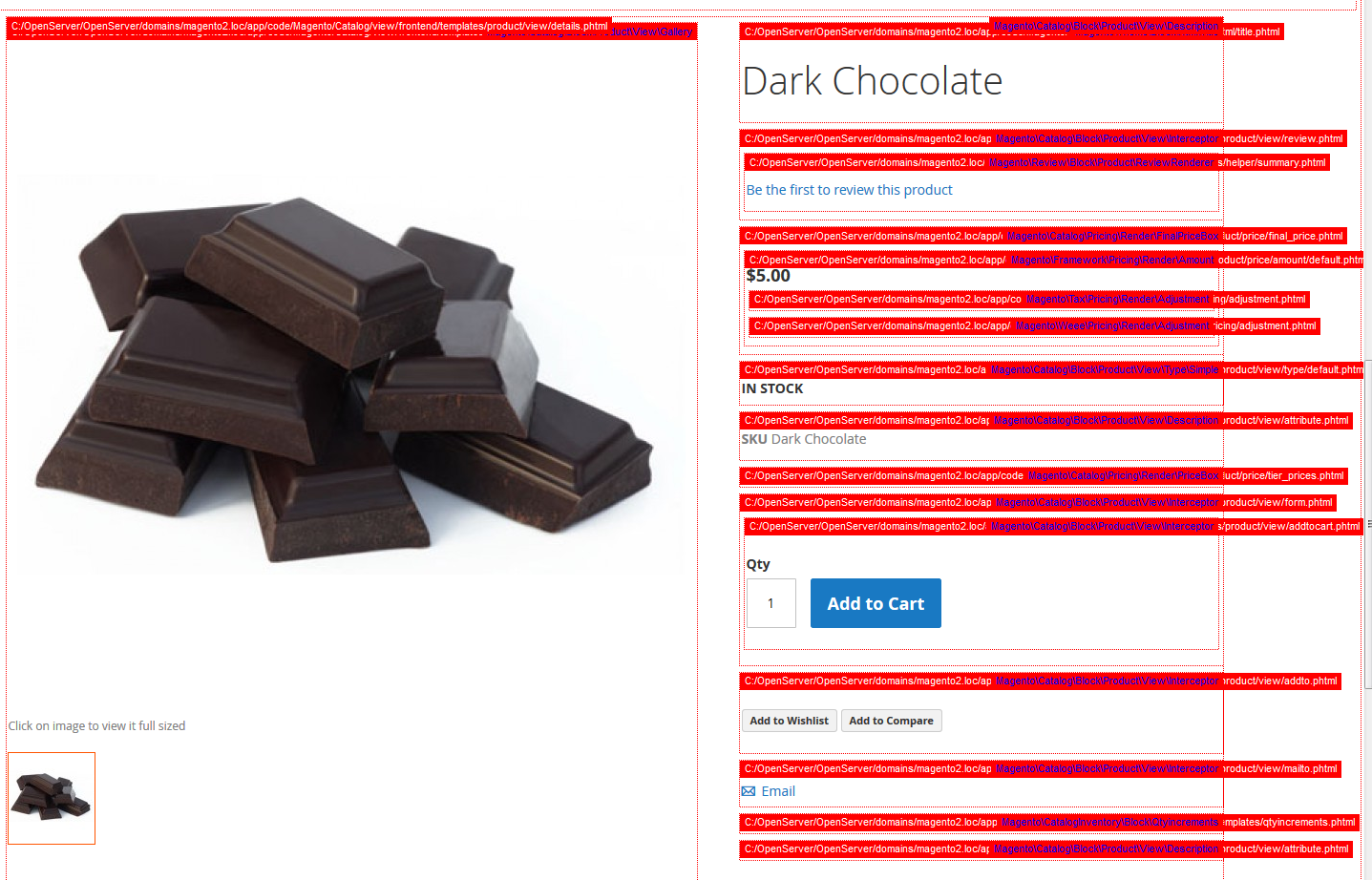
Để bật gợi ý template:

1. Nhấp vào **Cửa hàng** > **Cấu hình** > ADVANCED> **Nhà phát triển** .
2. Trong danh sách thả xuống **Phạm vi** ở góc trên bên trái, chọn chế độ xem mà bạn muốn gợi ý template.
3. Trong tab **Gỡ lỗi** , đặt **Mẹo đường dẫn template cho mặt tiền cửa hàng** thành **Có** . Để bật gợi ý đường dẫn cho Quản trị, hãy đặt **gợi ý đường dẫn template cho quản trị viên**thành **Có** .
4. Để lưu các thay đổi, hãy nhấp vào **Lưu cấu hình** ở góc trên bên phải.



Bây giờ bạn đã bật gợi ý template, hãy tải lại trang mà bạn muốn sửa đổi và xem lại đường dẫn cho file template hoặc file mà các gợi ý template hiển thị.

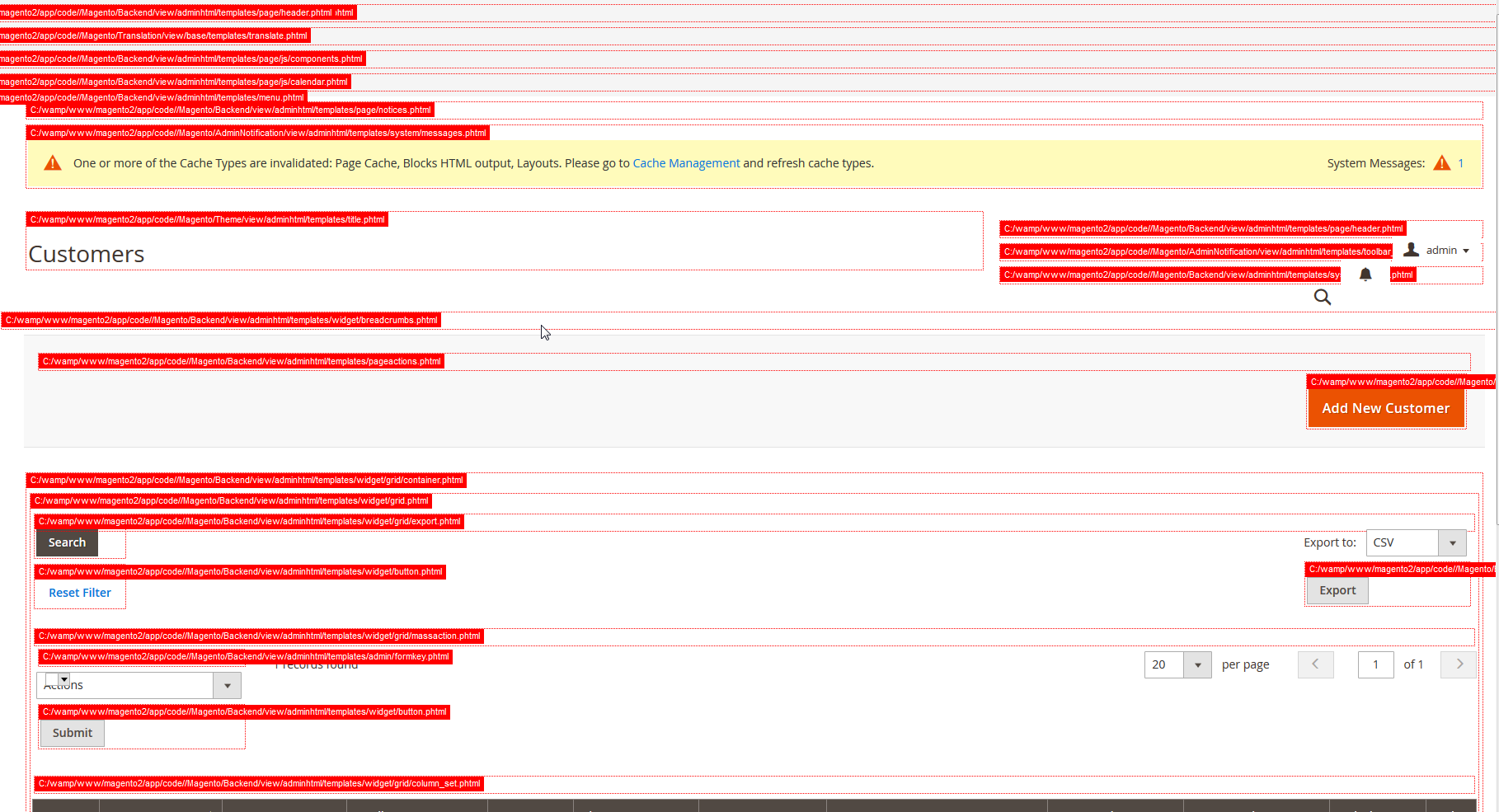
Ví dụ: dưới đây là cách trang danh mục cửa hàng trông giống như gợi ý template đã bật:



Trong ví dụ này, phần tử trang giỏ hàng nhỏ được xác định bởi <Magento\_Checkout\_module\_dir>/view/frontend/templates/cart/minicart.phtmltemplate:

(tên template nằm phía trên phần tử)

Dưới đây là cách trang Khách hàng hiển thị với gợi ý template được bật trong Quản trị viên:



Ngoài ra, bạn có thể thực hiện tìm kiếm văn bản trong hệ thống file bằng cách sử dụng tiêu đề được tạo hệ thống, tên lớp CSS , tiêu đề chặn, nhãn hoặc văn bản liên kết làm cụm từ tìm kiếm. Ví dụ, bằng cách sử dụng một công cụ gỡ lỗi trình duyệt, bạn có thể định nghĩa rằng lớp minicart css là minicart-wrapper.



Tìm kiếm thông qua thư mục ứng dụng cho các lần xuất hiện của “minicart-wrapper” trong .phtmlcác file trả về app/code/Magento/Checkout/view/frontend/templates/cart/minicart.phtmltemplate.

Vì không nên chỉnh sửa các file mặc định, bạn cần phải thêm các file ghi đè nếu bạn muốn tùy chỉnh template. Để biết chi tiết về các template ghi đè, vui lòng tham khảo [Tùy chỉnh template theme](https://devdocs.magento.com/guides/v2.2/frontend-dev-guide/templates/template-walkthrough.html) .

### Định vị bố cục

Cũng giống như các template, bố cục được lưu trên cơ sở từng mô-đun. Bạn có thể dễ dàng xác định vị trí file bố trí bằng cách xác định mô-đun template cho phần tử bạn quan tâm. Để định vị template, bạn có thể sử dụng Template gợi ý hoặc tìm kiếm văn bản trong thư mục ứng dụng, như được mô tả trước đây.

Sau khi bạn đã xác định mô-đun, bạn có thể tìm kiếm bố cục ở các vị trí sau:

1. <current\_theme\_dir>/<Namespace>\_<Module>/layout/
2. <parent\_theme(s)\_dir>/<Namespace>\_<Module>/layout/
3. <module\_dir>/view/frontend/layout/
4. <module\_dir>/view/base/layout/

Không có thuật toán đơn giản làm thế nào để xác định cùng một lúc các tập tin bố trí chính xác, nhưng trong hầu hết các trường hợp tên tập tin bố trí là tự mô tả. Ngoài ra, bạn có thể tìm kiếm chúng để đề cập đến các template tương ứng.

Thí dụ:

Giả sử bạn cần định vị bố cục chịu trách nhiệm hiển thị giỏ hàng nhỏ trên mặt tiền cửa hàng, khi theme Trống của Magento được áp dụng cho chế độ xem cửa hàng .

Sử dụng các gợi ý template chúng tôi xác định rằng template là app/code/Magento/Checkout/view/frontend/templates/cart/minicart.phtml, và trong đường dẫn, chúng ta thấy rằng nó thuộc về Magento\_Checkoutmô-đun.

Hãy tìm bố cục theo sơ đồ dự phòng:

1. Kiểm tra app/design/frontend/Magento/blank/Magento\_Checkout/bố cục. Để định vị bố trí được yêu cầu, hãy tìm thư mục này cho các lần xuất hiện của tên template, "minicart.phtml". Không tìm thấy file phù hợp nào, vì vậy chúng tôi tiến hành cấp dự phòng tiếp theo, đây là bố cục theme gốc.
2. Chúng tôi có thể tìm thấy thông tin về theme gốc trong file cấu hình theme theme.xml, tên theme chính được chỉ định ở đó trong <parent></parent>nút. Trong app/design/frontend/Magento/blank/theme.xmlkhông có <parent>nút nào, có nghĩa là theme Trống không có cha mẹ. Vì vậy, chúng ta nên tìm kiếm trên cấp độ dự phòng tiếp theo là bố cục mô-đun.
3. Các bố cục Magento\_Checkout được đặt tại app/code/Magento/Checkout/view/frontend/layout/. Sau khi tìm kiếm thư mục này cho các lần xuất hiện của “ minicart.phtml”, chúng tôi xác định rằng bố cục chúng tôi đang tìm kiếm là app/code/Magento/Checkout/view/frontend/layout/default.xml.

Sau khi bạn định vị file bố cục cần thiết, bạn có thể tạo file bố cục tùy chỉnh của mình với tên tương ứng trong thư mục theme của bạn để thêm nội dung [mở rộng](https://devdocs.magento.com/guides/v2.2/frontend-dev-guide/layouts/layout-extend.html) hoặc ghi [đè](https://devdocs.magento.com/guides/v2.2/frontend-dev-guide/layouts/layout-override.html) . Vui lòng xem [Tùy chỉnh bố cục theme](https://devdocs.magento.com/guides/v2.2/frontend-dev-guide/layouts/layout-overview.html) để biết thêm chi tiết.

### Định vị kiểu

Để xác định quy tắc CSS được áp dụng cho một phần tử nhất định, hãy tìm template cho trang chứa phần tử đó. Hoặc bạn có thể sử dụng các công cụ gỡ lỗi trình duyệt, để định vị tên lớp. Sau khi bạn tìm thấy tên lớp, hãy sử dụng tìm kiếm văn bản trong thư mục kiểu theme và mô-đun để định vị file .lesshoặc .cssfile định nghĩa lớp đó. Thực hiện tìm kiếm theo sơ đồ dự phòng sau:

1. Kiểu theme <current\_theme\_dir>/web/css/
2. Kiểu theme mô-đun <current\_theme\_dir>/<Namespace>\_<Module>/web/css/
3. Kiểu theme gốc <parent\_theme\_dir>/web/css/
4. Theme của phụ huynh <parent\_theme\_dir>/<Namespace>\_<Module>/web/css/
5. Kiểu mô-đun cho frontendkhu vực<module\_dir>/view/frontend/web/css/
6. Kiểu mô-đun cho basekhu vực<module\_dir>/view/base/web/css/

Thí dụ:

Hãy tìm tập tin định nghĩa trên các lớp CSS được sử dụng để hiển thị giỏ hàng nhỏ trên mặt tiền cửa hàng, khi theme Trống của Magento được áp dụng cho chế độ xem cửa hàng.

Trong template giỏ hàng mini, app/code/Magento/Checkout/view/frontend/templates/cart/minicart.phtmlphần tử cấp cao nhất có minicart-wrapperlớp.

Vì vậy, hãy tìm kiếm các lần xuất hiện của “ minicart-wrapper” theo kế hoạch dự phòng:

1. Tìm kiếm trong app/design/frontend/Magento/blank/web/css, tìm kiếm sẽ không trả về kết quả nào.
2. Tìm kiếm trong app/design/frontend/Magento/blank/Magento\_Checkout/web/css. minicart-wrapperKiểu "" được định nghĩa trongapp/design/frontend/Magento/blank/Magento\_Checkout/web/css/source/module/\_minicart.less

Sau khi bạn xác định file .csshoặc .lessfile nào định nghĩa lớp, bạn có thể ghi đè định nghĩa lớp mặc định trong tùy chỉnh .csshoặc .lessfile của mình.

## Cấu trúc theme Magento

Một [theme thiết kế](https://devdocs.magento.com/guides/v2.2/frontend-dev-guide/themes/theme-general.html#theme-gen-overview) là một phần quan trọng của ứng dụng Magento. Theme này mô tả cấu trúc tập tin của một theme Magento .

### Vị trí theme Magento

Theme cửa hàng được đặt ở vị trí thông thường app/design/frontend/<Vendor>/. Mặc dù về mặt kỹ thuật, chúng có thể nằm trong các thư mục khác. Ví dụ, Magento được xây dựng trong các theme có thể được đặt dưới vendor/magento/theme-frontend-<theme\_code>khi một cá thể Magento được triển khai từ kho lưu trữ Composer .

Mỗi theme phải được lưu trữ trong một thư mục riêng biệt:

app/design/frontend/<Vendor>/

├── <theme1>

├── <theme2>/

├── <theme3>

├--...

### Các thành phần theme

Cấu trúc của thư mục theme Magento thường giống như sau:

<theme\_dir>/

├── <Vendor>\_<Module>/

│ ├── web/

│ │ ├── css/

│ │ │ ├── source/

│ ├── layout/

│ │ ├── override/

│ ├── templates/

├── etc/

├── i18n/

├── media/

├── web/

│ ├── css/

│ │ ├── source/

│ ├── fonts/

│ ├── images/

│ ├── js/

├── composer.json

├── registration.php

├── theme.xml

Chúng ta hãy xem xét kỹ hơn từng thư mục con cụ thể.

Các cấu trúc thư mục và file được mô tả bên dưới là cấu trúc mở rộng nhất. Nó có thể không trùng với cấu trúc của cửa hàng của bạn.

|  |  |  |
| --- | --- | --- |
| **Danh mục** | **Cần thiết** | **Sự miêu tả** |
| /<Vendor>\_<Module> | không bắt buộc | Kiểu, bố cục và template cụ thể theo mô-đun. |
| /<Vendor>\_<Module>/web/css/source | không bắt buộc | Kiểu mô-đun cụ thể ( .cssvà / hoặc .lessfile). Các kiểu chung cho mô-đun nằm trong \_module.lessfile và các kiểu cho tiện ích đang ở trong đó \_widgets.less. |
| /<Vendor>\_<Module>/layout | không bắt buộc | Bố cục file mở rộng mô-đun mặc định hoặc bố cục theme gốc. |
| /<Vendor>\_<Module>/layout/override/base | không bắt buộc | Bố cục ghi đè bố cục mô-đun mặc định. |
| /<Vendor>\_<Module>/layout/override/<parent\_theme> | không bắt buộc | Bố cục ghi đè bố cục theme gốc cho mô-đun. |
| /<Vendor>\_<Module>/templates | không bắt buộc | Thư mục này chứa các template theme ghi đè các template mô-đun mặc định hoặc các template theme chính cho mô-đun này. Các template tùy chỉnh cũng được lưu trữ trong thư mục này. |
| /etc/view.xml | cần thiết cho một theme, nhưng tùy chọn nếu tồn tại trong theme gốc | File này chứa cấu hình hình ảnh cho tất cả hình ảnh và hình thu nhỏ của sản phẩm mặt tiền cửa hàng. |
| /i18n | không bắt buộc | file .csv có bản dịch. |
| /media | cần thiết | Thư mục này chứa bản xem trước theme (ảnh chụp màn hình theme của bạn). |
| /web | không bắt buộc | Các file tĩnh có thể được tải trực tiếp từ giao diện người dùng. |
| /web/css/source | không bắt buộc | Thư mục này chứa less các file cấu hình theme gọi các mixin cho các phần tử toàn cầu từ thư viện giao diện người dùng Magento và theme.less file ghi đè các giá trị biến mặc định. |
| /web/css/source/lib | không bắt buộc | Xem file ghi đè file thư viện UI được lưu trữ trong lib/web/css/source/lib |
| /web/fonts | không bắt buộc | Phông chữ theme. |
| /web/images | không bắt buộc | Hình ảnh được sử dụng trong theme này. |
| /web/js | không bắt buộc | Các file JavaScript theme. |
| /composer.json | không bắt buộc | Mô tả các phụ thuộc theme và một số thông tin meta. Sẽ ở đây nếu theme của bạn là một gói Composer. |
| /registration.php | cần thiết | Cần thiết để đăng ký theme của bạn trong hệ thống. |
| /theme.xml | cần thiết | File là bắt buộc vì nó tuyên bố một theme là một thành phần hệ thống. Nó chứa thông tin meta cơ bản, như tên theme và tên theme gốc, nếu theme được kế thừa từ một theme hiện có. Các tập tin được sử dụng bởi hệ thống Magento để nhận ra theme. |

### File theme

Ngoài file cấu hình và file siêu dữ liệu theme , tất cả các file theme đều nằm trong hai danh mục sau:

* File xem tĩnh
* File chế độ xem động

### File xem tĩnh

Một tập hợp các file theme được máy chủ trả về cho trình duyệt, không có bất kỳ xử lý nào, được gọi là các file tĩnh của một theme.

Các file tĩnh có thể được đặt trong thư mục theme như sau:

<theme\_dir>/

├── media/

├── web

│ ├── css/ (except the &quot;source&quot; sub-directory)

│ ├── fonts/

│ ├── images/

│ ├── js/

Sự khác biệt chính giữa file tĩnh và các file theme khác là file tĩnh xuất hiện trên trang web dưới dạng tham chiếu đến file, trong khi các file theme khác tham gia vào việc tạo trang nhưng không được tham chiếu rõ ràng trên trang web dưới dạng file.

Các file xem tĩnh có thể được truy cập bằng liên kết trực tiếp từ mặt tiền cửa hàng, được phân biệt là file theme công khai.

Thực sự có thể truy cập được đối với các file tĩnh công khai của trình duyệt được [xuất bản](https://devdocs.magento.com/guides/v2.2/config-guide/cli/config-cli-subcommands-static-view.html#config-cli-static-overview) vào /pub/static/frontend/<Vendor>/<theme>/<language>/css/thư mục.

### File chế độ xem động

Xem các tập tin được xử lý hoặc thực thi bởi máy chủ để cung cấp kết quả cho máy khách. Đây là: .lessfile, template và bố cục.

Các file xem động được đặt trong thư mục theme như sau:

<theme\_dir>/

├── Magento\_<module>/

│ ├── web/

│ │ ├── css/

│ │ │ ├── source/

│ ├── layout/

│ │ ├── override/

│ ├── templates/

├── web/

│ ├── css/

│ │ ├── source/

## Thừa kế theme

Theme kế thừa cho phép bạn dễ dàng mở rộng các theme và giảm thiểu các nỗ lực bảo trì. Bạn có thể sử dụng theme hiện có làm cơ sở cho các tùy chỉnh hoặc cập nhật thiết kế cửa hàng nhỏ, như trang trí ngày lễ. Thay vì sao chép các file theme mở rộng và sửa đổi những gì bạn muốn thay đổi, bạn có thể thêm ghi đè và mở rộng file.

Mức độ thừa kế theme không bị giới hạn.

Sự kế thừa theme dựa trên cơ chế dự phòng, đảm bảo rằng nếu một tập tin xem không được tìm thấy trong theme hiện tại, hệ thống sẽ tìm kiếm trong các theme tổ tiên, các tập tin mô-đun xem hoặc thư viện .

Thứ tự dự phòng hơi khác nhau đối với các tài sản tĩnh (CSS, JavaScript, phông chữ và hình ảnh) và các file theme khác, bố cục và template. Bài viết mô tả dự phòng cho từng loại file theme và cung cấp tổng quan về cách ghi đè theme tổ tiên và thiết kế mô-đun.

Để biết thông tin toàn diện về việc phát triển các thành phần theme, hãy xem các chương tiếp theo trong hướng dẫn này.

### Đặt theme gốc

Một theme cha mẹ được chỉ định trong theme.xmltập tin khai báo theme con .

Ví dụ: Theme Orange của OrangeCo kế thừa từ theme Magento Blank. Thừa kế được khai báo app/design/frontend/OrangeCo/orange/theme.xmlnhư sau:

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

<title>Orange</title>

<parent>Magento/blank</parent>

<media>

<preview\_image>media/preview.jpg</preview\_image>

</media>

</theme>

Bố mẹ và theme con có thể thuộc về các nhà cung cấp khác nhau. Ví dụ, theme tùy chỉnh của bạn có thể kế thừa từ theme Magento Blank.

### Ghi đè file view.xml

Nếu theme của bạn không chứa view.xml file cấu hình, nó sẽ được kế thừa từ theme gốc. Nếu bạn thêm <theme\_dir>/etc/view.xml file vào theme của mình, file sẽ ghi đè file của cha mẹ.

### Ghi đè nội dung tĩnh

Tài sản tĩnh hoặc các file chế độ xem tĩnh là các kiểu, JavaScript, hình ảnh và phông chữ.

Để tùy chỉnh các file xem tĩnh được xác định trong theme gốc, dạng xem mô-đun hoặc file thư viện, bạn có thể ghi đè chúng bằng cách thêm file có cùng tên trong vị trí có liên quan theo các sơ đồ dự phòng được mô tả thêm. Điều này cũng đề cập đến các .lesstập tin, mà về mặt kỹ thuật không phải là tài sản tĩnh.

Các thư mục cụ thể, nơi hệ thống tìm kiếm trong quá trình dự phòng, phụ thuộc vào ngữ cảnh mô-đun có được biết đến với file hay không. Sau đây là mô tả của cả hai tùy chọn.

Nếu ngữ cảnh mô-đun không được xác định cho một file:

1. Các file tĩnh theme hiện tại cho một miền địa phương cụ thể (ngôn ngữ được đặt cho mặt tiền cửa hàng):<theme\_dir>/web/i18n/<locale>
2. Các file tĩnh theme hiện tại: <theme\_dir>/web/
3. File tĩnh của tổ tiên, đệ quy, cho đến khi một theme không có cha mẹ đạt được:
   * <parent\_theme\_dir>/web/i18n/<locale>
   * <parent\_theme\_dir>/web/
4. Thư viện file xem tĩnh: lib/web/

Nếu ngữ cảnh mô-đun được xác định cho một file:

1. Theme hiện tại và các file tĩnh mô-đun miền địa phương hiện tại:<theme\_dir>/web/i18n/<locale>/<Namespace>\_<Module>
2. Các file tĩnh mô-đun theme hiện tại <theme\_dir>/<Namespace>\_<Module>/web/. Thí dụ:app/design/frontend/OrangeCorp/orange/Magento\_Catalog/web/
3. Mô-đun theme tổ tiên các file tĩnh, đệ quy, cho đến khi một theme không có tổ tiên được đạt tới:
   * <parent\_theme\_dir>/web/i18n/<locale>/<Namespace>\_<Module>
   * <parent\_theme\_dir>/<Namespace>\_<Module>/web/
4. Mô-đun xem tĩnh mô-đun cho frontendkhu vực:<module\_dir>/view/frontend/web/
5. Mô-đun xem tĩnh mô-đun cho basekhu vực:<module\_dir>/view/base/web/

**Thí dụ**

Một công ty tên OrangeCo đã tạo ra một theme có tên là Orange. Các tập tin theme được đặt tại app/design/frontend/OrangeCo/orange. Cam kế thừa từ theme Magento Blank.

Hãy tưởng tượng OrangeCo cần thêm một số trang trí ngày lễ mùa đông. Vì vậy, nó tạo ra một orange\_wintertheme mới , được kế thừa từ Orange. Theme nằm trong app/design/frontend/OrangeCo/orange\_winter.

Trong theme màu da cam có một hình nền footer nằm tại app/design/frontend/OrangeCo/orange/web/images/background.jpg.



OrangeCo muốn nó được thay thế bằng một kỳ nghỉ, vì vậy nó đặt một hình nền mới với chính xác cùng tên và phần mở rộngtrong app/design/frontend/OrangeCo/orange\_winter/web/images/background.jpg

Sau khi [áp dụng](https://devdocs.magento.com/guides/v2.2/frontend-dev-guide/themes/theme-apply.html) theme Mùa đông cam , hình ảnh kỳ nghỉ mới sẽ ghi đè hình ảnh từ màu cam, do đó, trên mặt tiền cửa hàng , nền nghỉ lễ sẽ hiển thị.



### Ghi đè template

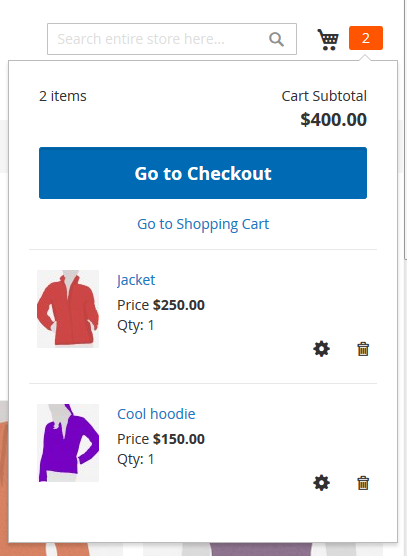
Đề án dự phòng cho các template là sau (ngữ cảnh mô-đun luôn được biết cho chúng):

1. Các template theme hiện tại: <theme\_dir>/<Namespace>\_<Module>/templates
2. Template theme tổ tiên, đệ quy, cho đến khi một theme không có tổ tiên được đạt tới: <parent\_theme\_dir>/<Namespace>\_<Module>/templates
3. Template mô-đun: <module\_dir>/view/frontend/templates

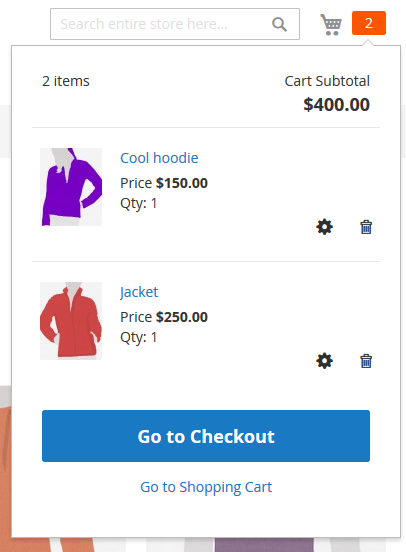
Vì vậy, nếu bạn cần tùy chỉnh một template nhất định, bạn cần phải tạo một template có trọng số có cùng tên trong ../templates/<path\_to\_template>thư mục trong các file mô-đun theme. Trong trường hợp <path\_to\_template>là đường dẫn đến template gốc.

Ví dụ, nếu bạn phải ghi đè lên <Magento\_Catalog\_module\_dir>/view/frontend/templates/category/widget/link/link\_block.phtmltemplate, <path\_to\_template>làcategory/widget/link/

**Ví dụ** Theo mặc định, theo template mô-đun, trong các sản phẩm giỏ mua hàng nhỏ được liệt kê trong nút Đi tới Thanh toán :



Thứ tự được xác định trong <Magento\_Checkout\_module\_dir>/view/frontend/templates/cart/minicart.phtmltemplate mô-đun. Theme Trống không ghi đè template này. OrangeCo quyết định họ muốn danh sách sản phẩm được hiển thị trước khi vào nút Đi tới Thanh toán. Để thực hiện việc này, họ cần phải thêm template ghi đè cho mô-đun tương ứng trong thư mục theme Màu cam: app/design/frontend/OrangeCo/orange/Magento\_Checkout/templates/cart/minicart.phtml Lưu ý rằng đường dẫn đến template bên trong templatesthư mục trong theme tương ứng với mô-đun đó trong mô-đun. Sau khi thay đổi thứ tự hoặc các phần tử trong các template, OrangeCo có giao diện đơn giản như sau:



Bạn có thể tìm ra những thay đổi mã chính xác được yêu cầu để thực hiện điều này và các tác vụ khác trong [theme Minh họa tùy chỉnh template](https://devdocs.magento.com/guides/v2.2/frontend-dev-guide/templates/template-sample.html) .

### Mở rộng bố cục

Cơ chế xử lý bố cục không liên quan đến dự phòng. Hệ thống thu thập các file bố trí theo thứ tự sau:

1. Bố cục theme hiện tại: <theme\_dir>/<Vendor>\_<Module>/layout/
2. Bố cục theme tổ tiên, bắt đầu từ tổ tiên xa nhất, đệ quy cho đến khi một theme không có cha mẹ đạt được: <parent\_theme\_dir>/<Vendor>\_<Module>/layout/
3. Bố cục mô-đun cho frontendkhu vực:<module\_dir>/view/frontend/layout/
4. Bố cục mô-đun cho basekhu vực:<module\_dir>/view/base/layout/

Không giống như các template hoặc hình ảnh, bố cục có thể không chỉ được ghi đè mà còn được mở rộng. Và cách được khuyến nghị để tùy chỉnh bố cục là mở rộng nó bằng cách tạo các file bố cục mở rộng theme.

Để thêm file bố cục mở rộng:

* Đặt file bố cục tùy chỉnh của bạn trong <theme\_dir>/<Vendor>\_<Module>/layout/thư mục.

**Thí dụ**

OrangeCo quyết định họ nên xóa liên kết “Báo cáo lỗi” khỏi chân trang, được xác định trong <Magento\_Theme\_module\_dir>/view/frontend/layout/default.xml Để thực hiện việc này, họ đã thêm bố cục mở rộng trong app/design/frontend/OrangeCo/orange/Magento\_Theme/layout/default.xml:

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

<body>

<referenceBlock name="report.bugs" remove="true"/>

</body>

</page>

Để biết thêm thông tin về việc mở rộng bố cục, hãy tham khảo bài viết [Mở rộng bố cục](https://devdocs.magento.com/guides/v2.2/frontend-dev-guide/layouts/layout-extend.html) .

### Ghi đè bố cục

Mặc dù không nên đề xuất bố cục ghi đè, nhưng vẫn có thể là một giải pháp cho một số tác vụ tùy chỉnh nhất định. Để ghi đè các hướng dẫn từ file bố cục theme tổ tiên:

* Tạo file bố cục có cùng tên trong <theme\_dir>/<Vendor>\_<Module>/layout/override/theme/<Vendor>/<ancestor\_theme>thư mục.

Để ghi đè hướng dẫn bố cục mô-đun (bố cục cơ sở):

* Tạo file bố cục có cùng tên trong <theme\_dir>/<Vendor>\_<Module>/layout/override/basethư mục.

## Thêm favicon tùy chỉnh

### Tổng quan chung

Magento cung cấp một favicon 16px x 16px mặc định mà bạn có thể ghi đè bằng cách tải lên biểu tượng tùy chỉnh trong Quản trị Magento hoặc bằng cách thêm nó theo cách thủ công vào một vị trí cụ thể trong thư mục theme . Nếu cả hai favicon đều tồn tại, cái bạn đã tải lên trong Quản trị sẽ được ưu tiên.

Nếu bạn muốn có favicon có kích thước khác nhau, bạn cần phải thêm chúng theo cách thủ công trong hệ thống file và xác định trong bố cục .

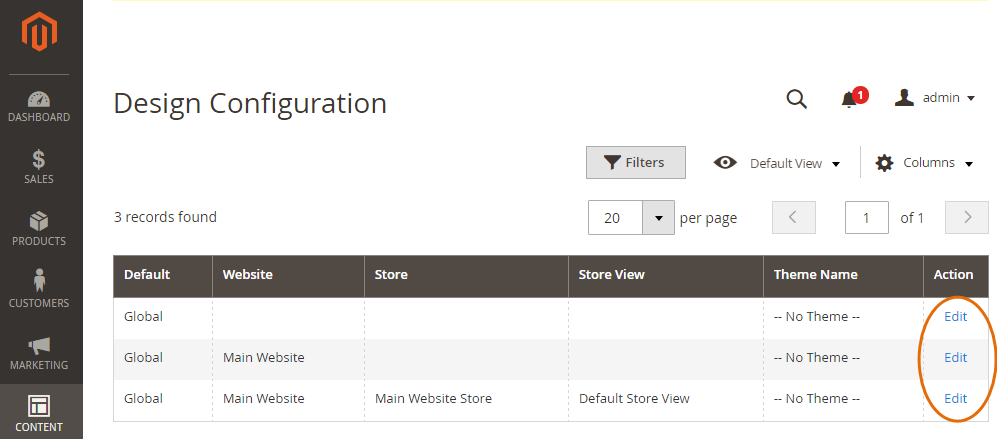
Magento hỗ trợ các loại tập tin sau đây cho favicon: .ico, .png, .gif, .jpg, .jpeg, .apng, .svg. Không phải tất cả các trình duyệt đều hỗ trợ tất cả các định dạng này. Các định dạng tập tin được hỗ trợ rộng rãi nhất để sử dụng cho một favicon là .ico.

Xem các phần sau để biết chi tiết về cách thêm favicon.

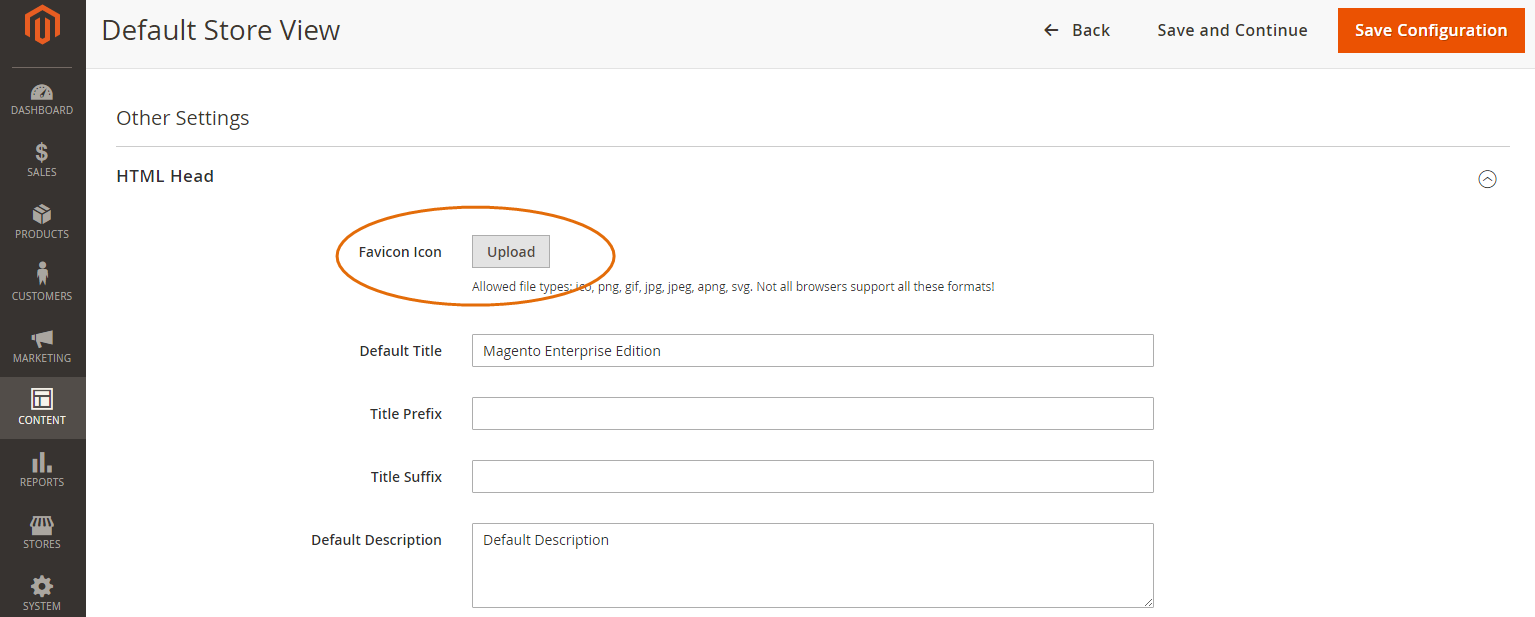
### Thêm một favicon tùy chỉnh trong Admin

Để thêm một favicon tùy chỉnh trong Magento Admin, hãy làm như sau:

1. Điều hướng đến **CONTENT** > **Thiết kế** > **Cấu hình** .
2. Trong lưới phạm vi, hãy quyết định cấp độ nào bạn sẽ định cấu hình favicon và nhấp vào **Chỉnh sửa** trong hàng tương ứng.



1. Trong tiêu đề **Cài đặt khác** , mở rộng tùy chọn **Đầu HTML** .
2. Bên cạnh **Biểu tượng Favicon** , nhấp vào **Tải lên** và chọn file.



1. Nhấp vào **Lưu cấu hình** ở góc trên bên phải để lưu các thay đổi.

Nếu bộ nhớ đệm được bật trong Quản trị viên của bạn, bạn sẽ nhận được thông báo làm mới một số loại bộ nhớ cache nhất định . Nhấp vào liên kết được cung cấp trong thông báo và sau đó nhấp vào **Flush Magento Cache** .

### Thêm favicon tùy chỉnh theo cách thủ công

Để ghi đè thủ công favicon 16px x 16px mặc định theo cách thủ công, hãy thêm tùy chỉnh của bạn favicon.icovào <your\_theme\_dir>/Magento\_Theme/web/thư mục.

Để thêm biểu tượng favicon có kích thước khác, hãy thực hiện các bước sau:

1. Thêm các biểu tượng của bạn vào <your\_theme\_dir>/Magento\_Theme/web/thư mục.
2. Trong <your\_theme\_dir>/Magento\_Theme/layout/default\_head\_blocks.xmlfile bố cục chỉ định đường dẫn đến các biểu tượng và kích thước của chúng.

Ví dụ: nếu bạn đã thêm favicon-32x32.pngbiểu tượng và muốn biểu tượng đó được sử dụng làm hình đại diện 32px x 32px, bạn default\_head\_blocks.xmlsẽ như sau:

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

<head>

<link src="Magento\_Theme::favicon-32x32.png" rel="icon" sizes="32x32" />

</head>

</page>

Để các thay đổi của bạn được áp dụng, hãy xóa bộ nhớ cache của trình duyệt và các thư mục sau trên máy chủ (không xóa .htaccessfile!):

* pub/static
* tất cả các thư mục dưới var

## Định cấu hình các thuộc tính hình ảnh cho một theme

Các thuộc tính của hình ảnh sản phẩm được sử dụng trên mặt tiền cửa hàng được lưu trữ trong view.xmlfile cấu hình. Theme này cung cấp tất cả chi tiết về những thuộc tính nào có sẵn và cách định cấu hình chúng.

Các thuộc tính cho hình ảnh được hiển thị trên các trang sản phẩm được xác định bởi các tùy chọn tiện ích của thư viện. Các tùy chọn của widget cũng có thể được cấu hình trong theme view.xml . Để biết thêm chi tiết, hãy xem theme [tiện ích Thư viện](https://devdocs.magento.com/guides/v2.2/javascript-dev-guide/widgets/widget_gallery.html) .

### Định cấu hình các thuộc tính hình ảnh trong view.xml

Vị trí thông thường của view.xmlmột theme là:

<theme\_dir>/etc/view.xml

Ví dụ, đây là view.xmlcủa theme Trống Magento: [app/design/frontend/Magento/blank/etc/view.xml](https://github.com/magento/magento2/blob/2.0/app/design/frontend/Magento/blank/etc/view.xml).

Trong view.xml, thuộc tính hình ảnh được cấu hình trong phạm vi của <images module="Magento\_Catalog">phần tử:

<images module="Magento\_Catalog">

...

<images/>

Thuộc tính hình ảnh được cấu hình cho từng loại hình ảnh được định nghĩa bởi idvà typethuộc tính của <image>phần tử:

<images module="Magento\_Catalog">

<image id="unique\_image\_id" type="image\_type">

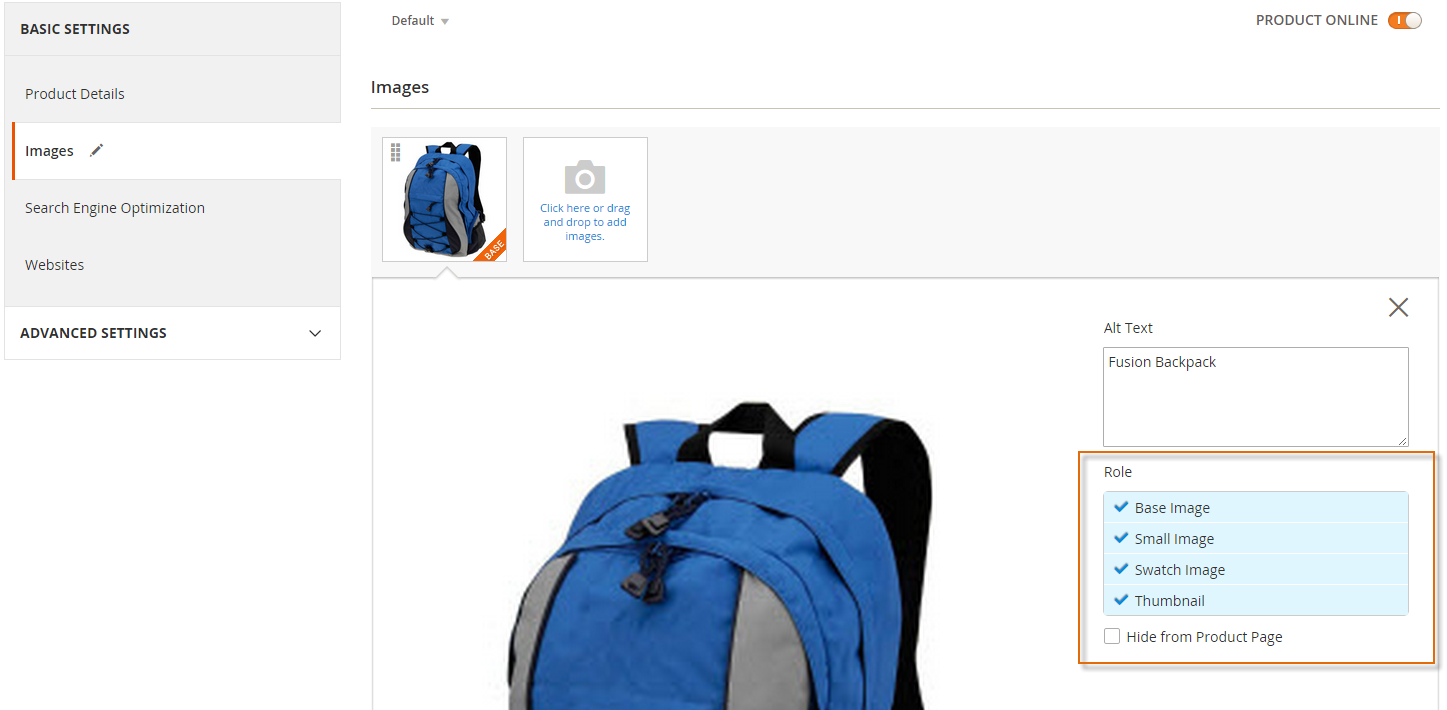
...

</image>

<images/>

Bảng sau mô tả chi tiết các thuộc tính:

|  |  |  |
| --- | --- | --- |
| **Thuộc tính** | **Kiểu** | **Sự miêu tả** |
| id | chuỗi | Định danh hình ảnh. Duy nhất trong phạm vi theme.  Có thể có bất kỳ giá trị nào, nhưng trong các theme của Magento idngoài ý nghĩa thì có ý nghĩa và mô tả vị trí của một hình ảnh.  Ví dụ: idgiá trị cho hình ảnh của sản phẩm bán kèm được hiển thị trong giỏ hàng là cart\_cross\_sell\_products.  idđược sử dụng trong các .phtmltemplate để xác định loại và thuộc tính của hình ảnh được hiển thị trong từng vị trí cụ thể trên một trang cụ thể. |
| type | chuỗi | Loại hình ảnh được xác định theo quy định id. Giá trị được phép:   * image - tương ứng với vai trò Image Base trong Magento Admin * small\_image - tương ứng với vai trò Hình ảnh Nhỏ trong Quản trị Magento * swatch\_image - tương ứng với vai trò Swatch Image trong Magento Admin * swatch\_thumb- tương ứng với vai trò Swatch Image trong Magento Admin . * thumbnail - tương ứng với vai trò Hình thu nhỏ trong Quản trị Magento |

Hình dưới đây minh họa cách vai trò hình ảnh cho hình ảnh sản phẩm được chỉ định trong Magento Admin: 

Các thuộc tính hình ảnh được xác định bởi các phần tử tương ứng, ví dụ:

<images module="Magento\_Catalog">

<image id="unique\_image\_id" type="image">

<width>100</width> <!-- Image width in px -->

<height>100</height> <!-- Image height in px -->

</image>

</images>

Bảng sau đây chứa danh sách tất cả các thuộc tính có thể được cấu hình:

|  |  |  |  |
| --- | --- | --- | --- |
| **Thành phần** | **Kiểu** | **Sự miêu tả** | **Cần thiết** |
| width | số nguyên | Chiều rộng hình ảnh tính theo pixel. | Không bắt buộc |
| height | số nguyên | Chiều cao hình ảnh tính theo pixel. | Không bắt buộc |
| constrain | boolean | Nếu được đặt thành true, hình ảnh nhỏ hơn yêu cầu của cấu hình sẽ không được phóng to. Giá trị mặc định: true. | Không bắt buộc |
| aspect\_ratio | boolean | Nếu được đặt thành true, tỷ lệ hình ảnh sẽ không thay đổi ngay cả khi được cấu hình yêu cầu. Giá trị mặc định: true. | Không bắt buộc |
| frame | boolean | Nếu được đặt thành true, ảnh sẽ không bị cắt. Giá trị mặc định: true. Chỉ áp dụng nếuaspect\_ratiođược đặt thành true. | Không bắt buộc |
| transparency | boolean | Nếu được đặt thành true, nền trong suốt của hình ảnh được lưu. Nếu được đặt thành false, hình ảnh sẽ có nền trắng (theo mặc định). Bạn có thể đặt màu cho nền bằng cách sử dụng background tham số. Giá trị mặc định: true. | Không bắt buộc |
| background | chuỗi | Màu cho nền hình ảnh. Không áp dụng cho hình ảnh có độ trong suốt, nếu transparencyđược đặt thành true. Định dạng: "[,,]", ví dụ: "[255, 255, 255]". | Không bắt buộc |

#### Đổi kích thước hình ảnh danh mục

Nói chung, hình ảnh sản phẩm được lưu trong bộ nhớ cache trong khi lưu sản phẩm. Tuy nhiên, magento catalog:images:resizelệnh cho phép bạn thay đổi kích thước tất cả hình ảnh để hiển thị trên mặt tiền cửa hàng của bạn. Các tình huống mà điều này có thể cần thiết có thể là:

* Sau khi bạn nhập sản phẩm, có thể có hình ảnh có kích thước khác nhau
* Nếu hình ảnh được thay đổi kích thước hoặc xóa theo cách thủ công từ bộ nhớ cache

Mỗi hình ảnh được gán cho một sản phẩm phải được thay đổi kích thước theo siêu dữ liệu hình ảnh được xác định trong [view.xml](https://devdocs.magento.com/guides/v2.2/frontend-dev-guide/themes/theme-create.html#fedg_create_theme_how-to-images)file cấu hình của mô-đun . Sau khi thay đổi kích thước một hình ảnh, bản sao được thay đổi kích thước của nó được lưu trữ trong bộ nhớ cache ( /pub/media/catalog/product/cachethư mục). Magento phục vụ hình ảnh mặt tiền cửa hàng từ bộ nhớ cache.

Sử dụng lệnh:

php <magento install dir>/bin/magento catalog:images:resize

Lệnh này không có đối số hoặc tùy chọn. Chỉ báo tiến trình hiển thị khi lệnh chạy.

Thông báo Product images resized successfullyhiển thị để xác nhận lệnh đã thành công.

## Định cấu hình video sản phẩm

Trong Magento 2 trên trang sản phẩm, bạn có thể thêm video từ các tài nguyên bên ngoài (hiện tại, từ [YouTube](https://youtube.com/) và [Vimeo](https://vimeo.com/) ). Video được [thêm vào Quản trị](http://docs.magento.com/m2/2.0/ee/user_guide/catalog/product-video.html?Highlight=product%20video) khi tạo hoặc chỉnh sửa sản phẩm. Một số tùy chọn video sản phẩm nhất định có thể được đặt trong config.xmlfile cấu hình. Các cài đặt này không phải là theme cụ thể.

### Định cấu hình tùy chọn video sản phẩm

Bạn có thể đặt các tùy chọn video sản phẩm sau:

|  |  |  |  |
| --- | --- | --- | --- |
| **Tùy chọn** | **Sự miêu tả** | **Kiểu** | **Mặc định** |
| play\_if\_base | Phát tự động khi tải trang. | Boolean | 0  (video không được phát khi tải trang) |
| show\_related | Hiển thị video có liên quan. | Boolean | 0  (video liên quan không được hiển thị) |
| video\_auto\_restart | Tự động phát lại video. | Boolean | 0  (video không tự động được phát lại) |

Các tùy chọn được đặt trong mô-đunconfig.xml tùy chỉnh của bạn .

Thí dụ:

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

<default>

<catalog>

<product\_video>

<play\_if\_base>1</play\_if\_base>

<show\_related>1</show\_related>

<video\_auto\_restart>1</video\_auto\_restart>

</product\_video>

</catalog>

</default>

</config>

Vì lợi ích của khả năng tương thích, nâng cấp và bảo trì dễ dàng, không chỉnh sửa mã Magento mặc định. Thay vào đó, hãy thêm các tùy chỉnh của bạn vào một mô-đun riêng biệt.

## Ghép JavaScript

Gói JavaScript là một kỹ thuật tối ưu hóa mà bạn có thể sử dụng để giảm số lượng yêu cầu máy chủ cho các file JavaScript. Việc kết hợp hoàn thành việc này bằng cách hợp nhất nhiều file JavaScript với nhau thành một file để giảm số lượng yêu cầu trang.

### Bật tính năng đóng gói JavaScript

Magento phải được đặt thành [chế độ sản xuất] để có thể hoạt động.

Để kích hoạt gói JavaScript:

1. Đăng nhập vào khu vực quản trị
2. Hướng đến Stores -> Configuration -> Advanced -> Developer -> Javascript Settings
3. Đặt Enable Javascript Bundlingthành Yes.

Để tối ưu hóa nhóm, hãy đặt các cài đặt sau thành Yes:

* Javascript Settings -> Merge JavaScript Files
* Javascript Settings -> Minify Javascript Files
* Static Files Settings -> Sign Static Files

### Cách gói hoạt động trong Magento

Theo mặc định, Magento gói tất cả các file JavaScript và tải chúng đồng bộ trên mọi lần tải trang. Điều này có nghĩa là mỗi gói được nạp một cái khác và gây ra sự chậm trễ cho các hành vi JavaScript trên một trang.

#### Loại trừ file

Các <exclude>mục nhập trong một etc/view.xmltập tin của theme nói với Magento những tập tin nó không nên bó. Các file JavaScript bị loại trừ khỏi gói được tải không đồng bộ bởi RequireJS khi cần.

Không gộp các file JavaScript được sử dụng để thử nghiệm hoặc phát triển vì các file này sẽ được tải trên mỗi trang.

Đoạn mã sau từ [theme Luma của Magento](https://github.com/magento/magento2/blob/2.2.3/app/design/frontend/Magento/luma/etc/view.xml#L270) hiển thị các loại file bạn nên loại trừ trong theme của mình.

#### Đặt kích thước file bó

Các bundle\_sizeentry kiểm soát kích thước tập tin của các bó được tạo ra. Đặt giá trị này thành số lớn sẽ giảm số lượng gói được tạo nhưng kích thước file sẽ lớn. Một số nhỏ sẽ tăng số lượng gói được tạo, nhưng kích thước file sẽ nhỏ hơn.

Bạn muốn cân bằng số lượng file để tải xuống trên trang có kích thước dữ liệu. Một kết nối duy nhất nên được khoảng 100kb.

### Tinh chỉnh theme của bạn

Có nhiều cách để điều chỉnh etc/view.xmlfile của theme của bạn .

Ví dụ, theme Luma của Magento được cấu hình để hoạt động tốt cho tất cả các trang, nhưng bằng cách thêm hoặc xóa các mục bên trong exclude, bạn có thể tối đa hóa hiệu suất trình duyệt cho trang chủ, danh mục hoặc trang sản phẩm.

Làm theo các bước sau để giúp bạn xác định file JavaScript nào sẽ được nhóm cho theme của bạn:

1. Tạo một trang trống với các bố trí bạn muốn điều chỉnh.
2. So sánh các file JavaScript được tải trong các trang với các file JavaScript trong Magento.
3. Sử dụng kết quả của so sánh đó để tạo danh sách loại trừ của bạn.

# Template

## Tổng quan về template

Trong các template ứng dụng Magento là một phần của lớp xem. Template xác định chính xác cách nội dung của [khối bố cục](https://devdocs.magento.com/guides/v2.2/frontend-dev-guide/layouts/layout-overview.html) được trình bày trên một trang: thứ tự, các lớp CSS , nhóm phần tử, v.v. Trong hầu hết các trường hợp, các template không chứa bất kỳ logic nào về việc liệu chúng có hoặc sẽ không được trả lại hay không, điều này thường được xử lý bởi các file bố cục . Khi template được gọi trong bố cục, template sẽ được hiển thị.

Các template Magento mặc định là các file PHTML . Các template HTML cũng được sử dụng cho các kịch bản lệnh [Knockout JS](http://knockoutjs.com/index.html) .

## Hướng dẫn tùy chỉnh template

### Có gì trong theme này

Theme này hướng dẫn bạn cách tùy chỉnh template.

### Điều kiện tiên quyết

[Đặt](https://devdocs.magento.com/guides/v2.2/config-guide/cli/config-cli-subcommands-mode.html) ứng dụng Magento của bạn sang [chế độ](https://devdocs.magento.com/guides/v2.2/config-guide/bootstrap/magento-modes.html) nhà phát triển . Chế độ ứng dụng ảnh hưởng đến cách các file tĩnh được lưu trữ bởi Magento. Các khuyến nghị về phát triển theme mà chúng tôi cung cấp trong chương này là nhà phát triển / chế độ mặc định cụ thể.

### Hướng dẫn tùy chỉnh template

Để tùy chỉnh template:

1. Định vị template được liên kết với trang / khối bạn muốn thay đổi bằng cách sử dụng các [gợi ý template](https://devdocs.magento.com/guides/v2.2/frontend-dev-guide/themes/debug-theme.html#debug-theme-templ) .
2. Sao chép template vào thư mục theme của bạn theo [quy ước lưu trữ template](https://devdocs.magento.com/guides/v2.2/frontend-dev-guide/templates/template-override.html#template-convention) .
3. Thực hiện các thay đổi cần thiết.

Để thêm template mới trong một theme:

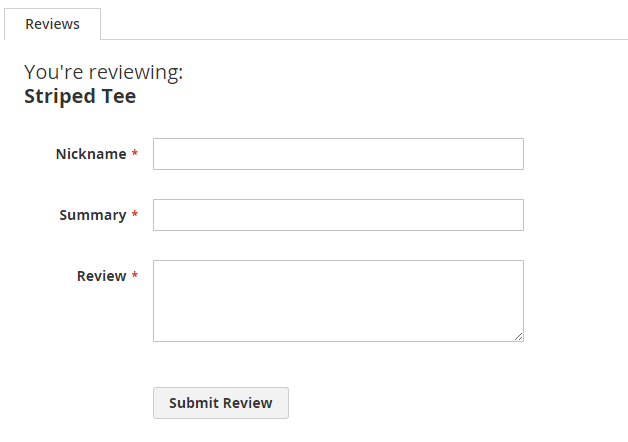
1. Thêm một template trong thư mục theme của bạn theo quy ước lưu trữ template.
2. Gán template của bạn cho một khối trong [file bố cục tương ứng](https://devdocs.magento.com/guides/v2.2/frontend-dev-guide/templates/template-override.html#template-layout) .

Nếu bạn thêm một .htmltemplate mới , và sau đó chỉnh sửa nó, các thay đổi sẽ không áp dụng cho đến khi bạn xóa tất cả các file trong pub/static/frontendvà các var/view\_preprocessedthư mục và tải lại các trang. Bạn có thể xóa các tập tin bằng tay hoặc chạy grunt clean:<theme\_name>lệnh trong CLI. Để biết chi tiết về cách sử dụng Grunt trong Magento, hãy xem [Cài đặt và cấu hình Grunt](https://devdocs.magento.com/guides/v2.2/frontend-dev-guide/css-topics/css_debug.html) .

### Hình minh họa hướng dẫn: thêm thông báo vào biểu template đánh giá của khách hàng

Một tùy chỉnh nhỏ để minh họa hướng dẫn: trong theme Orange của họ, công ty OrangeCo muốn thêm một văn bản ngắn vào template đánh giá sản phẩm để khuyến khích khách hàng viết đánh giá.

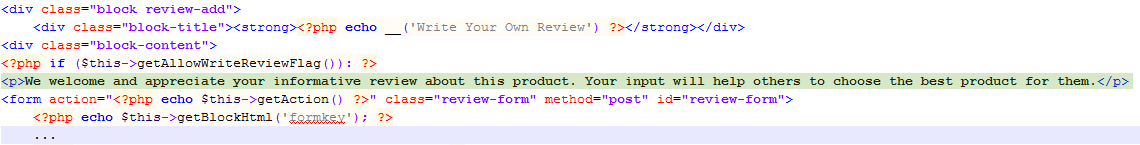
Hình ảnh sau đây minh họa cách biểu template đánh giá mặc định trông như thế nào:



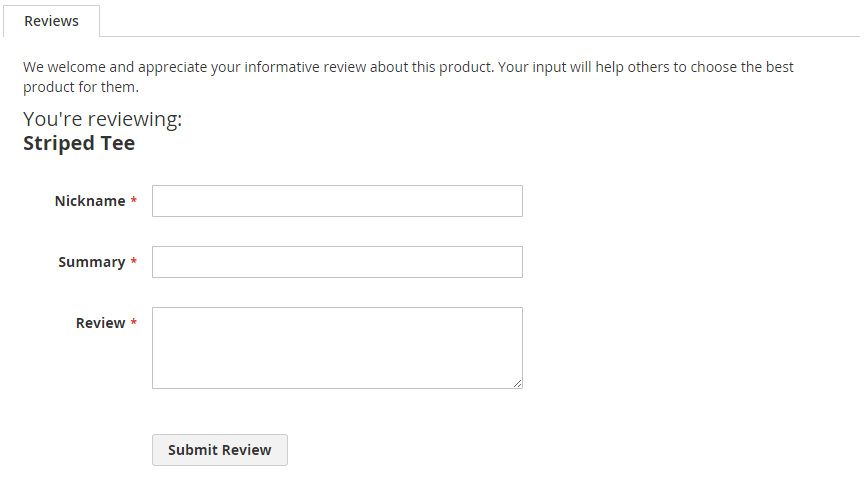
Để thêm văn bản, OrangeCo cần ghi đè template biểu template đánh giá mặc định trong theme Màu cam.

Thứ nhất, họ sao chép các form.phtmltemplate từ <Magento\_Review\_module\_dir>/view/frontend/templatesđến thư mục tương ứng trong thư mục theme Orange: app/design/frontend/OrangeCo/orange/Magento\_Review/templates.

Trong form.phtmlfile theme, họ thêm đoạn mã HTML với thông báo trước <form>:



Dưới đây là cách biểu template sẽ hiển thị khi theme màu cam được áp dụng trong cửa hàng:



## Template khái niệm cơ bản

### Có gì trong theme này

Theme này giải thích cách các template mặc định hoạt động trong ứng dụng Magento.

### Cách bắt đầu template

Các template được bắt đầu trong file bố cục và mỗi khối bố cục có template được liên kết.

Template được chỉ định trong templatethuộc tính của lệnh <block>bố cục.

Lấy ví dụ này từ [app/code/Magento/Catalog/view/frontend/layout/catalog\_category\_view.xml](https://github.com/magento/magento2/blob/2.0/app/code/Magento/Catalog/view/frontend/layout/catalog_category_view.xml):

<block class="Magento\Catalog\Block\Category\View" name="category.image" template="Magento\_Catalog::category/image.phtml">

Các category.imagekhối được đưa ra bởi các image.phtmltemplate trong categorythư mục con của các Magento\_Catalog mô-đun thư mục template.

Thư mục template Magento\_Cataloglà app/code/Magento/Catalog/view/frontend/templates.

### Vị trí template

Các template được lưu trữ ở các vị trí sau:

* Template mô-đun: <module\_dir>/view/frontend/templates/<path\_to\_templates>
* Template theme: <theme\_dir>/<Namespace>\_<Module>/templates/<path\_to\_templates>

<path\_to\_templates> cho biết không có hoặc nhiều cấp độ thư mục.

Ví dụ:

* app/code/Magento/Catalog/view/frontend/templates/product/widget/new/content/new\_grid.phtml
* app/code/Magento/Checkout/view/frontend/templates/cart.phtml

### Ghi đè template

Đối với các file template có cùng tên, các quy tắc ghi đè sau sẽ áp dụng:

* Các template theme ghi đè template mô-đun
* Template [theme con](https://devdocs.magento.com/guides/v2.2/frontend-dev-guide/themes/theme-inherit.html) ghi đè template theme mẹ

Để thay đổi đầu ra được xác định bởi template hiện có, hãy ghi đè template trong theme tùy chỉnh của bạn. Khái niệm này là cơ sở của việc tùy biến template trong Magento.

<Magento\_Theme\_module\_dir>/view/base/templates/root.phtmllà template gốc cho tất cả các trang mặt tiền cửa hàng trong ứng dụng Magento. File này có thể được ghi đè trong một theme giống như bất kỳ file template nào khác.

Không giống như các template khác, root.phtmlchứa các doctypeđặc tả và đóng góp vào <head>và <body>các phần của tất cả các trang được hiển thị bởi ứng dụng Magento.

### Lấy giá trị đối số từ bố cục

Giá trị đối số được đặt trong file bố cục được truy cập trong các template bằng cách sử dụng get{ArgumentName}()và has{ArgumentName}()phương thức.

Xem [hướng dẫn Bố cục](https://devdocs.magento.com/guides/v2.2/frontend-dev-guide/layouts/xml-instructions.html#getter)

### Sử dụng các thẻ ngắn PHP trong các file template PHTML

Các echolệnh trong PHP có thể được viết bằng cách sử dụng thẻ ngắn trong Magento templates.

Ví dụ:

<?= $block->getAdjustmentsHtml() ?>

giống như viết

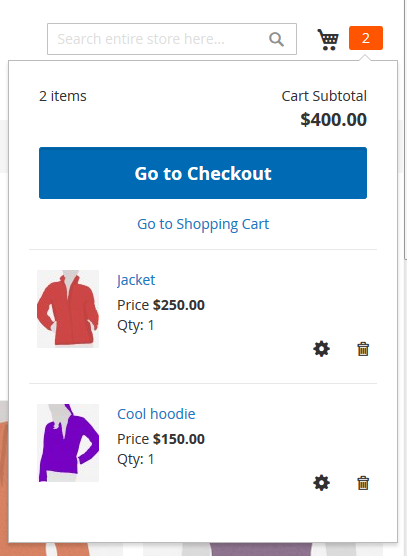
<?php echo $block->getAdjustmentsHtml() ?>

## Minh họa tùy chỉnh template

Theme này chứa minh họa từng bước giải quyết một tác vụ tùy chỉnh thiết kế điển hình bằng cách sử dụng các template.

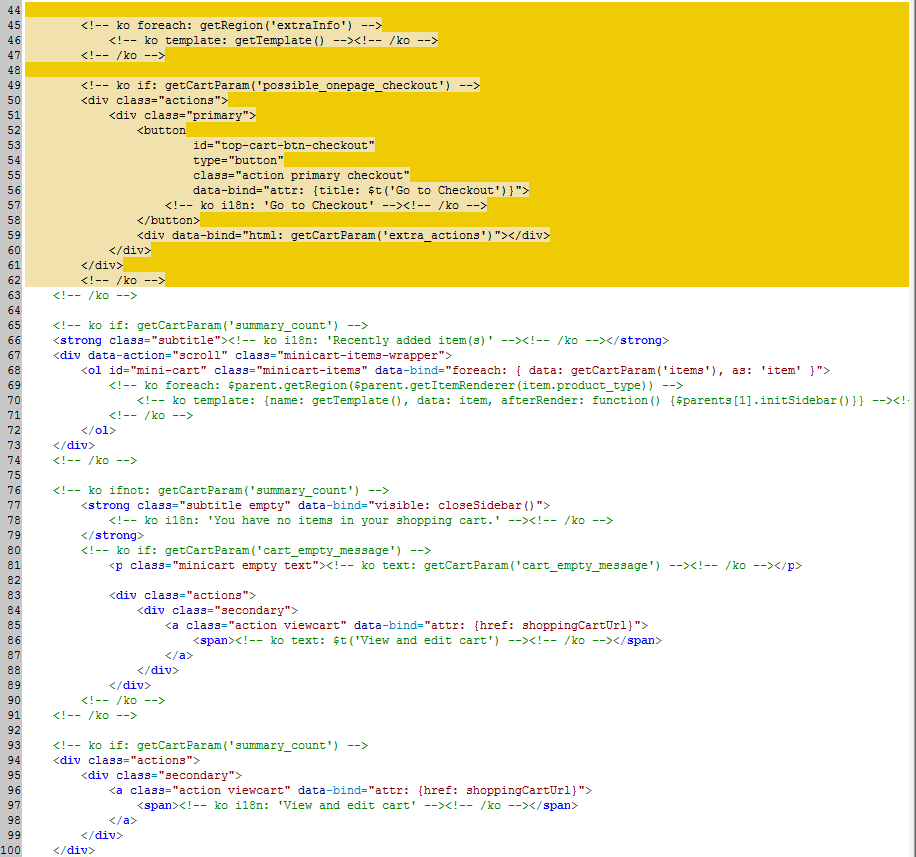
### Tùy chỉnh template template: thay đổi bố cục của giỏ hàng nhỏ

Trong theme Magento basic Blank, trong giỏ hàng nhỏ, các sản phẩm được liệt kê dưới nút **Đi tới Thanh toán** , như sau:



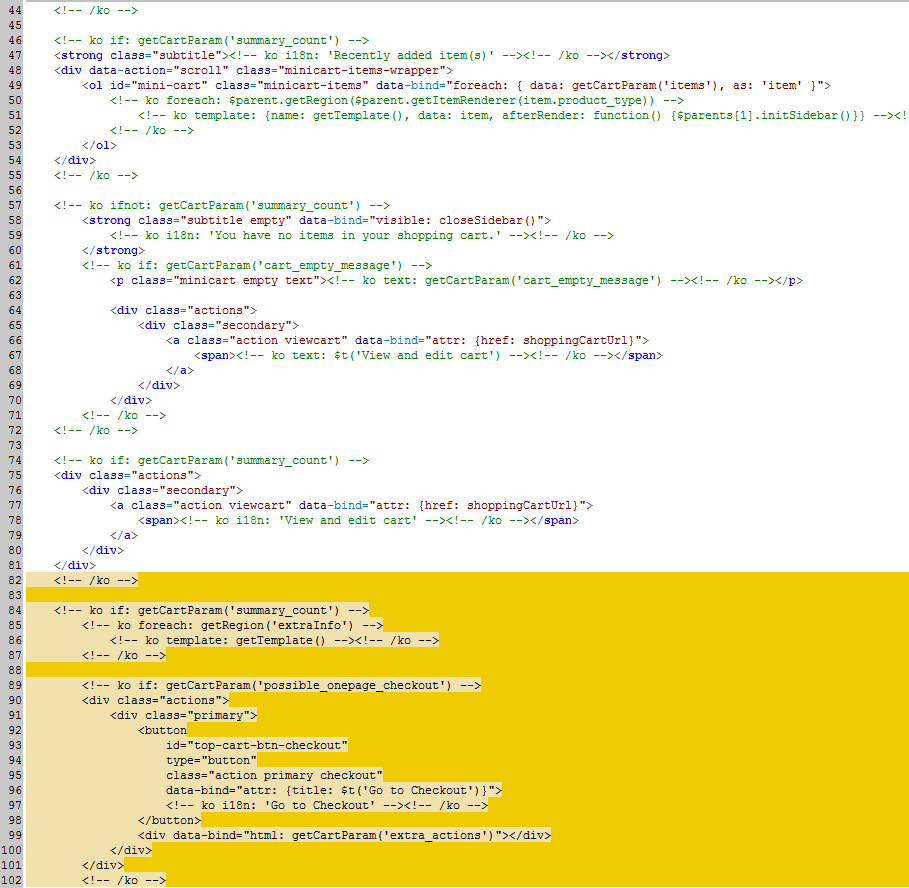
OrangeCo quyết định họ muốn thay đổi điều này và hiển thị danh sách sản phẩm trước khi **vào** nút **Đi đến Thanh toán** .

Template chịu trách nhiệm hiển thị các mục và điều khiển giỏ mua hàng mini là [<Magento\_Checkout\_module\_dir>/view/frontend/web/template/minicart/content.html](https://github.com/magento/magento2/blob/2.2/app/code/Magento/Checkout/view/frontend/web/template/minicart/content.html). Đây là một phần của mã OrangeCo đã làm việc với:

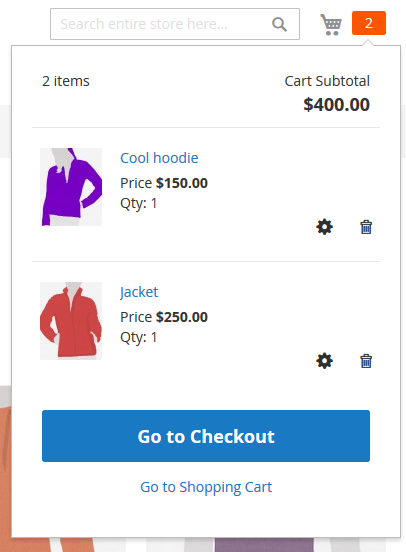


Họ đã tạo ra một theme mới Orange và sao chép content.htmlvào thư mục theme:app/design/frontend/OrangeCo/orange/Magento\_Checkout/web/template/minicart/content.html.

Trong bản sao của các template, họ đã thay đổi thứ tự của các khối như sau:



Khi theme màu da cam được áp dụng, giỏ hàng nhỏ với các sản phẩm trông giống như sau:



## Tùy chỉnh template email

### Tùy chỉnh template email

Các template email được lưu trữ trong <module\_dir>/view/<area>/emailthư mục của các mô-đun tương ứng. Ví dụ: template cho email giao dịch đơn đặt hàng mới cho mô-đun Bán hàng được đặt tại [<Magento\_Sales\_module\_dir>/view/frontend/email/order\_new.html](https://github.com/magento/magento2/blob/2.1/app/code/Magento/Sales/view/frontend/email/order_new.html).

Chúng tôi khuyên bạn không nên thay đổi các file Magento mặc định. Nếu bạn muốn tùy chỉnh các template mặc định, bạn nên tạo các template tùy chỉnh của mình và định cấu hình Magento để sử dụng chúng thay vì các template mặc định.

Bạn có thể thêm các template tùy chỉnh dưới dạng file vật lý trong theme tùy chỉnh của mình hoặc tạo chúng bằng cách sử dụng Quản trị viên Magento . Cả hai cách tiếp cận được mô tả trong các phần sau.

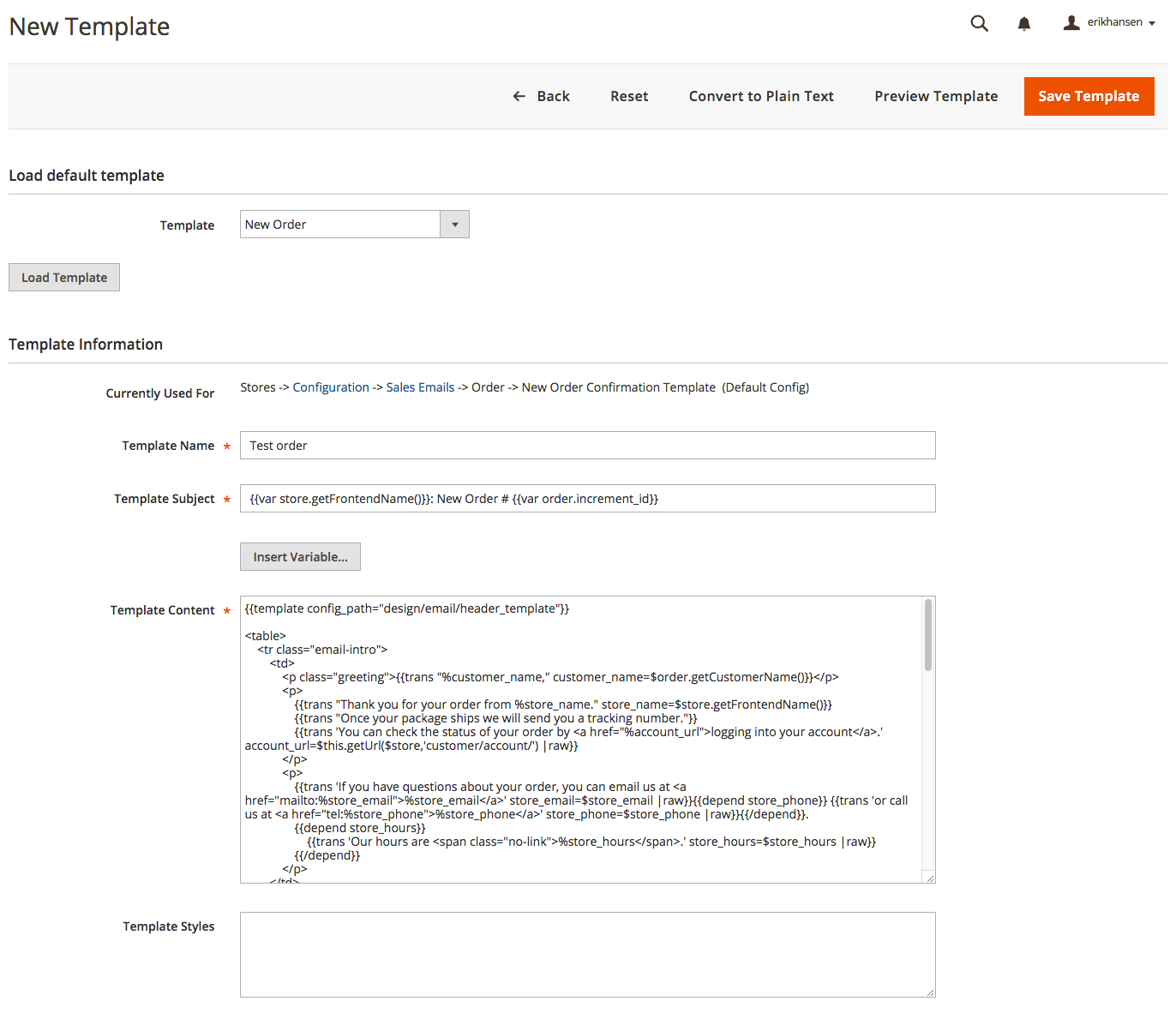
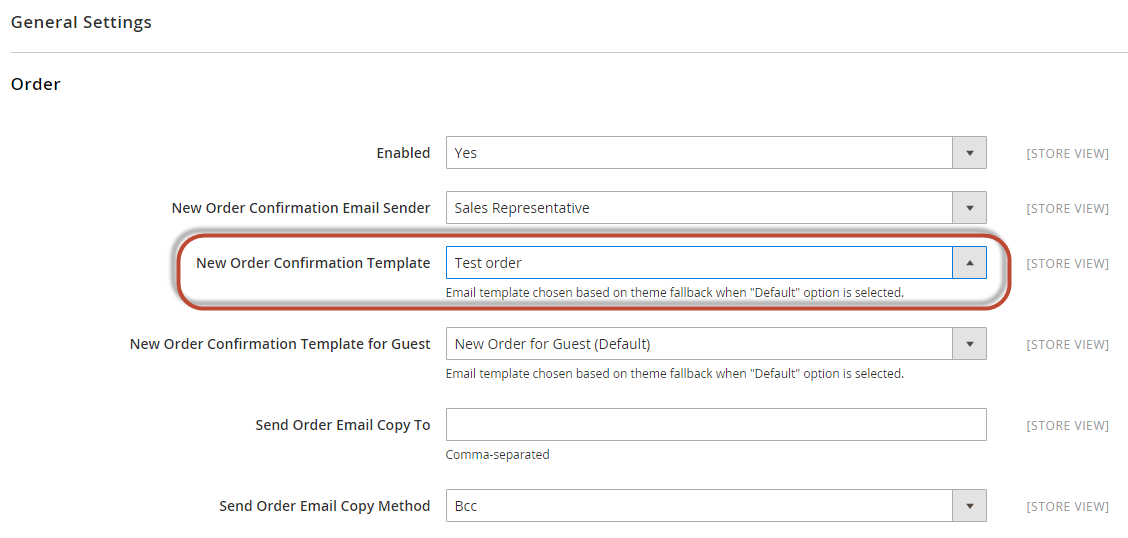
#### Tùy chỉnh template email bằng cách sử dụng một theme

Override template email bằng cách tạo ra các template trong một thư mục mới trong theme tùy chỉnh của bạn, sử dụng mô hình này: <theme\_dir>/<ModuleVendorName>\_<ModuleName>/email. Ví dụ: để ghi đè template email đặt hàng mới, hãy tạo template có tên order\_new.htmltrong <theme\_dir>/Magento\_Sales/emailthư mục.

[Dự phòng template](https://devdocs.magento.com/guides/v2.2/frontend-dev-guide/themes/theme-inherit.html#theme-inherit-templates) được hỗ trợ cho các template email, do đó, các theme chính của theme hiện tại của bạn được tìm kiếm template.

#### Tùy chỉnh các template email bằng cách sử dụng Quản trị Magento

Bất kỳ template nào được định cấu hình trong Quản trị viên Magento đều được ưu tiên hơn các template mặc định hoặc dựa trên theme.

1. Trong Quản trị Magento, điều hướng đến **MARKETING** > Truyền thông> **Template email**
2. Nhấp vào **Thêm template mới** .
3. Nếu bạn muốn sử dụng template mặc định làm điểm bắt đầu, trong phần **Tải template mặc định** , hãy chọn template và nhấp vào **Tải template** . Đường dẫn đến cài đặt cấu hình cho mỗi template mặc định sẽ hiển thị trong trường **Hiện đang được sử dụng** trong phần Thông tin template.   
   Hãy lưu ý đường dẫn này bởi vì bạn sẽ cần nó sau này khi bạn cấu hình template mới này sẽ được sử dụng thay cho template mặc định.   
   
4. Trong **Tên template** , hãy nhập tên để xác định template trong Quản trị Magento.
5. Trong **Theme template** , thêm văn bản thuần để sử dụng làm Theme của các email được gửi bằng template bạn tạo. Trường này có thể chứa các biến hệ thống.
6. Tùy chỉnh nội dung template. Để biết chi tiết, hãy xem [phần tùy chỉnh nội dung](https://devdocs.magento.com/guides/v2.2/frontend-dev-guide/templates/template-email.html#customize-content) .
7. Trong **Template Kiểu** , tùy chọn thêm kiểu CSS cho template. Các kiểu này được thêm vào bên trong <style>thẻ trong <head>email. Thông thường, bạn sẽ sử dụng [file LESS](https://devdocs.magento.com/guides/v2.2/frontend-dev-guide/templates/template-email.html#email-styles) để thực hiện thay đổi kiểu cho email vì một số ứng dụng email không hỗ trợ kiểu trong <style>thẻ.
8. Nhấp vào **Lưu template** .
9. Bây giờ bạn đã tạo template, bạn phải định cấu hình template đó để sử dụng:
   1. Nếu bạn chưa làm như vậy, hãy đăng nhập vào Magento Admin với tư cách quản trị viên.
   2. Nhấp vào **STORES** > Cài đặt> **Cấu hình** > SALES> **Email bán hàng** .
   3. Trong ngăn bên trái, tìm phần chứa template bạn muốn ghi đè. Đây là phần được tham chiếu bởi **Hiện đang sử dụng cho** trong template mới của bạn. (Xem bước 3 trước đó trong phần này.)   
      Ví dụ, nếu bạn đã tạo một template "Đơn hàng Mới", phần cấu hình là **Thứ tự** như hình dưới đây.   
      
   4. Chọn template mới được tạo của bạn từ danh sách.
   5. Nhấp vào **Lưu cấu hình** .

#### Tùy chỉnh template đầu trang và chân trang

Mỗi template email đầu vào bao gồm template đầu trang và chân trang bằng cách sử dụng hai chỉ thị này: {{template config\_path="design/email/header\_template"}}và {{template config\_path="design/email/footer\_template"}}. Theo mặc định, hai chỉ thị này tải nội dung từ các file này:

* [<Magento\_Email\_module\_dir>/view/frontend/email/header.html](https://github.com/magento/magento2/blob/2.1/app/code/Magento/Email/view/frontend/email/header.html)
* [<Magento\_Email\_module\_dir>/view/frontend/email/footer.html](https://github.com/magento/magento2/blob/2.1/app/code/Magento/Email/view/frontend/email/footer.html)

Bạn có thể tùy chỉnh template đầu trang và chân trang bằng cách sử dụng phương pháp tùy chỉnh [theme](https://devdocs.magento.com/guides/v2.2/frontend-dev-guide/templates/template-email.html#customize-email-theme) hoặc [quản trị được](https://devdocs.magento.com/guides/v2.2/frontend-dev-guide/templates/template-email.html#customize-email-admin) thảo luận trước đây.

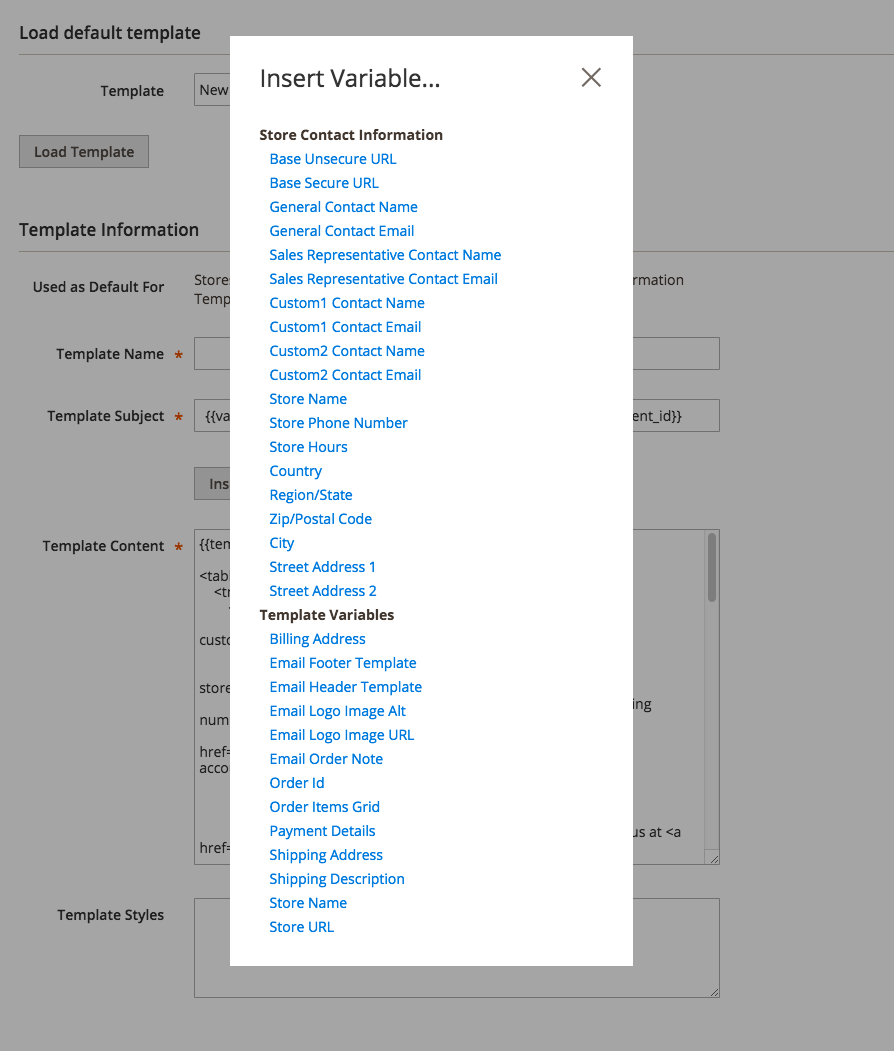
#### Tùy chỉnh nội dung email

Để thêm thông tin liên quan đến cửa hàng và bán hàng vào template, hãy sử dụng biến hệ thống.

Biến hệ thống là trình giữ chỗ được thay thế bằng các giá trị cụ thể khi email thực được tạo. Ví dụ: {{var store\_hours}}biến được thay thế bằng giá trị được đặt trong phần **STORES** > Cài đặt> **Cấu hình** > CHUNG> **Chung** > **Thông tin cửa hàng** .

Bạn cũng có thể tạo biến tùy chỉnh của riêng mình và đặt giá trị của chúng trong Quản trị viên, trong **SYSTEM** > **Biến tùy chỉnh** .

Để thêm biến vào nội dung template của bạn:

1. Trong Quản trị Magento, điều hướng đến **MARKETING** > Truyền thông> **Template email**
2. Tạo template mới hoặc chỉnh sửa template hiện có.
3. Nhấp để đặt con trỏ vào văn bản để chèn biến.
4. Nhấp vào **Chèn biến** . Cửa sổ bật lên chứa danh sách các biến mở ra, bao gồm các biến tùy chỉnh. Các biến trong **Thông tin liên hệ của cửa hàng** có sẵn trong tất cả các template email trong khi các biến trong phần **Biến template** cụ thể cho template bạn đang chỉnh sửa. Hình dưới đây cho thấy một ví dụ:  
   
5. Nhấp vào tên biến bắt buộc.   
   Mã biến được chèn vào nội dung template.

Việc chọn các biến có sẵn tùy thuộc vào template bạn sử dụng làm cơ sở. Các biến template cụ thể được chứa trong một <!--@vars @-->nhận xét ở đầu mỗi template trên hệ thống file. (Ví dụ: xem [ứng dụng / mã / Magento / Khách hàng / xem / giao diện người dùng / email / account\_new.html](https://github.com/magento/magento2/blob/2.1/app/code/Magento/Customer/view/frontend/email/account_new.html#L8) .

### Kiểu cho template email

Một số ứng dụng email (ví dụ: Gmail) chỉ hỗ trợ các kiểu CSS đã được áp dụng dưới dạng kiểu "nội tuyến" trên stylethuộc tính của thẻ HTML. Bởi vì điều này, phần lớn các kiểu email được áp dụng như kiểu nội tuyến. Kiểu nội tuyến được cung cấp bởi [Emogrifier](https://github.com/jjriv/emogrifier) thư viện Emogrifier, trong đó có mã HTML và CSS và cho biết thêm tất cả các phong cách CSS để stylethuộc tính của các thẻ HTML.

#### Kiểu nội tuyến

Các <Magento\_Email\_module\_dir>/view/frontend/email/header.htmltập tin có chứa một inlinecsschỉ thị:

{{inlinecss file="css/email-inline.css"}}

Các inlinecsschỉ thị cho Magento những file nào để áp dụng phong cách như inline trên template email.

Ví dụ, giả sử một email đang được gửi từ một cửa hàng được cấu hình với theme Magento Luma. Các inlinecsschỉ thị đầu tiên tìm kiếm một email-inline.lesstập tin trong <Magento\_Luma\_theme\_dir>/web/css/email-inline.less. Tuy nhiên, vì tập tin đó không tồn tại, nó sẽ rơi trở lại <Magento\_Blank\_theme\_dir>/web/css/email-inline.lesstập tin. Nội dung của tập tin đó sau đó sẽ được biên dịch và nội dung của nó được áp dụng như kiểu nội tuyến cho template email.

Hãy tham khảo [Emogrifier README](https://github.com/jjriv/emogrifier#supported-css-selectors%22) để xem những bộ chọn CSS nào được hỗ trợ.

#### Kiểu không phải nội tuyến

Phong cách không nội tuyến cho email đến từ các kiểu toàn cầu và template cụ thể, như được mô tả trong các phần sau.

#### Kiểu không nội dòng chung

Mặc dù phần lớn các kiểu phải được áp dụng nội tuyến, nhưng có một số kiểu CSS nhất định không thể áp dụng nội tuyến, chẳng hạn như truy vấn phương tiện hoặc :hoverkiểu giả. Các kiểu này phải nằm trong <style type="text/css"></style>thẻ để chúng hoạt động.

Các <Magento\_Email\_module\_dir>/view/frontend/email/header.htmltập tin có chứa một csschỉ thị bên trong một <style>thẻ:

<style type="text/css">

{{var template\_styles|raw}}

{{css file="css/email.css"}}

</style>

Các csschỉ thị biên dịch nội dung của các tập tin được cung cấp và kết quả đầu ra nó.

Ví dụ, giả sử một email đang được gửi từ một cửa hàng được cấu hình với theme Magento Luma. Các csschỉ thị đầu tiên tìm kiếm một email.lesstập tin trong <Magento\_Luma\_theme\_dir>/web/css. Tuy nhiên, vì tập tin không tồn tại ở đó, nó rơi trở lại <Magento\_Blank\_theme\_dir>/web/css/email.less. Nội dung của tập tin đó được biên dịch và nội dung đầu ra của nó trong <style>thẻ.

#### Kiểu không phải nội dòng dành riêng cho template

Như đã đề cập trong phần trước, header.htmlfile sẽ xuất ra {{var template\_styles|raw}}biến.

Giá trị của biến đó đến từ bất kỳ điều nào sau đây:

* Bất kỳ kiểu nào bạn thêm vào bất kỳ htmltemplate email nào bên trong một khối nhận xét, như trong ví dụ sau, được bao gồm trong template\_stylesbiến:
* <!--@styles
* .example-style { color: green; }
* @-->
* Nếu bạn tùy chỉnh email giao dịch bằng cách sử dụng Quản trị Magento, bạn có thể thêm kiểu CSS vào trường **Kiểu template** để bao gồm các kiểu đó trong template\_stylesbiến.

#### Cách sắp xếp kiểu email

Các kiểu email được chia thành nhiều file khác nhau.

|  |  |
| --- | --- |
| **Tập tin** | **Sự miêu tả** |
| /web/css/email.less</code></p> </td> | Nhập các file cần thiết và sau đó xuất các kiểu được đưa vào |

## Bảo mật template XSS

### Các biện pháp bảo mật chống lại các cuộc tấn công XSS

Để ngăn chặn các vấn đề [XSS,](https://en.wikipedia.org/wiki/Cross-site_scripting) Magento khuyến nghị các quy tắc sau đây để thoát khỏi đầu ra trong các template:

* Nếu một phương pháp chỉ ra rằng nội dung được bỏ trốn, không thoát khỏi: getTitleHtml(), getHtmlTitle()(tên đã sẵn sàng cho HTML đầu ra)
* Loại đúc và php chức năng count()không cần thoát (ví dụ echo (int)$var, echo (bool)$var, echo count($var))
* Đầu ra trong dấu nháy đơn không cần thoát (ví dụ echo 'some text')
* Đầu ra trong dấu ngoặc kép không có biến không cần thoát (ví dụ echo "some text")
* Đối với tất cả các trường hợp khác, hãy thoát dữ liệu bằng cách sử dụng [các chức năng thoát cụ thể](https://devdocs.magento.com/guides/v2.2/frontend-dev-guide/templates/template-security.html#escape-functions-for-templates) .

Template mã sau minh họa đầu ra XSS an toàn trong các template:

<?php echo $block->getTitleHtml() ?>

<?php echo $block->getHtmlTitle() ?>

<?php echo $block->escapeHtml($block->getTitle()) ?>

# <?php echo (int)$block->getId() ?>

<?php echo count($var); ?>

<?php echo 'some text' ?>

<?php echo "some text" ?>

<a href="<?php echo $block->escapeUrl($block->getUrl()) ?>"><?php echo $block->getAnchorTextHtml() ?></a>

#### Chức năng thoát cho template

Đối với các trường hợp đầu ra sau, hãy sử dụng hàm được chỉ định để tạo đầu ra XSS-safe.

**Trường hợp:** Đầu ra JSON   
**Chức năng:** Không cần hàm nào cho đầu ra JSON.

<!-- In this example $postData is a JSON string -->

<button class="action" data-post='<?php /\* @noEscape \*/ echo $postData ?>' />

**Trường hợp:** Đầu ra chuỗi không được chứa hàm HTML   
**:** escapeHtml

Bạn có thể chuyển vào một mảng thẻ được phép tùy chọn sẽ không bị thoát.

Nếu thẻ được cho phép, các thuộc tính sau đây sẽ không được thoát: id, class, href, target, stylevà title. Bất kỳ thuộc tính nào khác cho thẻ được cho phép đó sẽ được thoát.

embed, iframe, video, source, object, audion, scriptVà imgthẻ sẽ không được phép không phụ thuộc vào nội dung của mảng này.

Nếu văn bản của bạn chứa các ký tự đặc biệt, chúng phải được mã hóa dưới dạng thực thể HTML, chẳng hạn như <cho **<** hoặc >for **>** .

<span class="label"><?php echo $block->escapeHtml($block->getLabel()) ?></span>

// Escaping translation

<div id='my-element'><?php echo $block->escapeHtml(\_\_('Only registered users can write reviews. Please <a href="%1">Sign in</a> or <a href="%2">create an account</a>', $block->getLoginUrl(), $block->getCreateAccountUrl()), ['a']) ?></div>

**Trường hợp:** URL sản lượng   
**Chức năng:** escapeUrl

<a href="<?php echo $block->escapeUrl($block->getCategoryUrl()) ?>">Some Link</a>

<script>

var categoryUrl = '<?php echo $block->escapeJs($block->escapeUrl($block->getCategoryUrl())) ?>';

</script>

**Trường hợp:** Chuỗi bên trong JavaScript   
**Chức năng:** Trong ngữ cảnh JavaScript, hãy sử dụng escapeJshàm.

Trong trường hợp mã JavaScript xuất nội dung lên trang, hãy sử dụng escapeUrlhoặc escapeHtmlchức năng thích hợp.

Ví dụ: khi chuỗi đầu ra URL nằm trong ngữ cảnh JavaScript, hãy sử dụng cả hai escapeJsvà escapeUrl. Nếu bạn chèn chuỗi đầu ra từ bên trong một ngữ cảnh JavaScript vào DOM, hãy sử dụng cả hai escapeJsvà escapeHtml.

var field<?php echo $block->escapeJs($block->getFieldNamePostfix()) ?> = window.document.getElementById('my-element');

var categoryUrl = '<?php echo $block->escapeJs($block->escapeUrl($block->getCategoryUrl())) ?>';

// Escaping content that will be inserted into DOM

var string = <?php echo $block->escapeJs($block->escapeHtml(\_\_('Only registered users can write reviews. Please <a href="%1">Sign in</a> or <a href="%2">create an account</a>', $block->getLoginUrl(), $block->getCreateAccountUrl()), ['a'])) ?>

jQuery('#my-element').append(string);

// Here we are not inserting the translated string into the DOM, so it is ok if the string contains non-allowed tags or

// JavaScript because it will be handled as a string. Do not use escapeHtml here, the browser will display quotes

// and other symbols as HTML entities (&#039;, &quot;, &amp;, etc)

alert('<?php echo $block->escapeJs(\_\_('You are not authorized to perform this action.')) ?>');

**Trường hợp:** Các chuỗi bên trong thuộc tính HTML   
**Chức năng:** escapeHtmlAttr

<span class="<?php echo $block->escapeHtmlAttr($block->getSpanClass()) ?>">Product Description</span>

<input name="field" value="<?php echo $block->escapeHtmlAttr($block->getFieldValue()) ?>" />

<!-- Escaping translation inside attributes -->

<img src="product-blue.jpg" alt="<?php echo $block->escapeHtmlAttr(\_\_('A picture of the product in blue')) ?>" />

#### Kiểm tra tĩnh

Để kiểm tra template của bạn cho các lỗ hổng XSS, bạn có thể sử dụng thử nghiệm tĩnh XssPhtmlTemplateTest.phptrong dev\tests\static\testsuite\Magento\Test\Php\.

Thử nghiệm tĩnh này tìm tất cả các cuộc gọi echo trong các template PHTML và xác định xem đầu ra có được thoát đúng cách hay không.

Nó bao gồm các trường hợp sau:

* /\* @noEscape \*/trước khi xuất. Đầu ra không yêu cầu thoát. Kiểm tra là màu xanh lá cây.
* Các phương thức có chứa "html"trong tên của chúng (ví dụ echo $object->{suffix}Html{postfix}()). Dữ liệu đã sẵn sàng cho đầu ra HTML. Kiểm tra là màu xanh lá cây.
* Phương pháp AbstractBlock escapeHtml, escapeHtmlAttr, escapeUrl, escapeJsđược phép. Kiểm tra là màu xanh lá cây.
* Loại đúc và chức năng php count()được phép (ví dụ echo (int)$var, (bool)$var, count($var)). Kiểm tra là màu xanh lá cây.
* Đầu ra trong dấu nháy đơn (ví dụ echo 'some text'). Kiểm tra là màu xanh lá cây.
* Đầu ra trong dấu ngoặc kép không có biến (ví dụ echo "some text"). Kiểm tra là màu xanh lá cây.
* Khác đã đề cập trước đó. Đầu ra không được thoát. Thử nghiệm có màu đỏ.