# events

Các event được gửi đi bởi các module khi các hành động nhất định được kích hoạt. Ngoài các event riêng của nó, Magento cho phép bạn tạo ra các event của riêng bạn có thể được gửi đi trong mã của bạn. Khi một event được gửi đi, nó có thể truyền dữ liệu cho bất kỳ người quan sát nào được cấu hình để xem event đó.

events.xml

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

## Dispatching events

Event có thể được gửi đi bằng cách sử dụng class Magento\Frameword\Event\Manager. Class này có thể thu được thông tin thông qua [dependency injection](https://devdocs.magento.com/guides/v2.2/extension-dev-guide/depend-inj.html) bằng cách xác định vào phương thức khởi tạo \_\_contructor.

Để gửi một event, hãy gọi hàm dispatch() của class người quản lý event và cung cấp nó với tên của event mà bạn muốn gửi cùng với một mảng dữ liệu mà bạn muốn cung cấp cho người quan sát.

Ví dụ sau đây cho bạn thấy cách gửi một event có và không có một mảng dữ liệu

namespace MyCompany\MyModule;

use Magento\Framework\Event\ObserverInterface;

class MyClass

{

/\*\*

\* @var EventManager

\*/

private $eventManager;

public function \_\_construct(\Magento\Framework\Event\Manager $eventManager)

{

$this->eventManager = $eventManager;

}

public function something()

{

$eventData = null;

// Code...

$this->eventManager->dispatch('my\_module\_event\_before');

// More code that sets $eventData...

$this->eventManager->dispatch('my\_module\_event\_after',['myEventData'=>$eventData]);

}

}

## Tạo event mới

Các event tùy chỉnh có thể được gửi đi bằng cách đơn giản truyền tên event duy nhất cho trình quản lý event khi bạn gọi hàm dispatch(). Tên event duy nhất của bạn được tham chiếu trong events.xml của module bạn chỉ định người quan sát nào sẽ phản ứng với event đó.

## Khu vực event

Nói chung, vị trí của events.xml tệp sẽ nằm trong <module-root>/etc . Các nhà quan sát được liên kết với các event ở đây sẽ theo dõi các event này trên toàn cầu. Các events.xml cũng có thể được định nghĩa dưới <module-root>/etc/frontendvà <module-root>/etc/adminhtml thư mục để cấu hình quan sát để chỉ xem cho các event trong những lĩnh vực cụ thể.

# Observers

Các nhà quan sát là một loại Magento nhất định có thể ảnh hưởng đến hành vi chung, hiệu suất, hoặc thay đổi logic nghiệp vụ. Các nhà quan sát được thực hiện bất cứ khi nào event được cấu hình để xem được trình quản lý event gửi đi.

## Tạo observer

Để tạo một người quan sát, bạn phải đặt tệp lớp của bạn trong <module-root>/Observer mục của bạn . Lớp người quan sát của bạn nên triển khai [Magento\Framework\Event\ObserverInterface](https://github.com/magento/magento2/blob/2.0/lib/internal/Magento/Framework/Event/ObserverInterface.php) và xác định execute chức năng của nó .

Ví dụ

namespace MyCompany\MyModule\Observer;

use Magento\Framework\Event\ObserverInterface;

class MyObserver implements ObserverInterface

{

public function \_\_construct()

{

//Observer initialization code...

//You can use dependency injection to get any class this observer may need.

}

public function execute(\Magento\Framework\Event\Observer $observer)

{

//Observer execution code...

}

}

Một trong những tính năng mạnh mẽ hơn của các nhà quan sát là họ có thể sử dụng các tham số được truyền vào event khi nó được gửi đi. Dưới đây là ví dụ về người quan sát thu thập dữ liệu được truyền vào khi event được gửi đi.

namespace MyCompany\MyModule\Observer;

use Magento\Framework\Event\ObserverInterface;

class AnotherObserver implements ObserverInterface

{

public function \_\_construct()

{

//Observer initialization code...

//You can use dependency injection to get any class this observer may need.

}

public function execute(\Magento\Framework\Event\Observer $observer)

{

$myEventData = $observer->getData('myEventData');

//Additional observer execution code...

}

}

## Đăng ký event

obsever có thể được cấu hình để xem các event nhất định trong events.xmltập tin.

Phần tử observer xml có các thuộc tính sau:

* name (bắt buộc) - Tên của người quan sát cho định nghĩa event.
* instance (bắt buộc) - Tên lớp học đầy đủ của người quan sát.
* disabled- Xác định xem người quan sát này có đang hoạt động hay không. Giá trị mặc định là sai.
* shared- Xác định [lifestyle](https://devdocs.magento.com/guides/v2.2/extension-dev-guide/build/di-xml-file.html#object-lifestyle-configuration)  của lớp học. Mặc định là sai.

Dưới đây là ví dụ về cách chỉ định người quan sát để xem các event nhất định:

<?xml version="1.0"?>

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

<event name="my\_module\_event\_before">

<observer name="myObserverName" instance="MyCompany\MyModule\Observer\MyObserver" />

</event>

<event name="my\_module\_event\_after">

<observer name="myObserverName" instance="MyCompany\MyModule\Observer\AnotherObserver" />

</event>

</config>

1. List all event in magento 2

# Events in PHP Files

* Event name: sales\_quote\_remove\_item
* File: vendor/magento/Collect/Model/Quote.php

$this->\_eventManager->dispatch('sales\_quote\_remove\_item', ['quote\_item' => $item]);

* Event name: sales\_quote\_add\_item
* File: vendor/magento/Collect/Model/Quote.php

$this->\_eventManager->dispatch('sales\_quote\_add\_item', ['quote\_item' => $item]);

* Event name: sales\_quote\_product\_add\_after
* File: vendor/magento/Collect/Model/Quote.php

$this->\_eventManager->dispatch('sales\_quote\_product\_add\_after', ['items' => $items]);

* Event name: \_merge\_before
* File: vendor/magento/Collect/Model/Quote.php

$this->\_eventManager->dispatch(

$this->\_eventPrefix . '\_merge\_before',

[$this->\_eventObject => $this, 'source' => $quote]

);

* Event name: \_merge\_after
* File: vendor/magento/Collect/Model/Quote.php

$this->\_eventManager->dispatch(

$this->\_eventPrefix . '\_merge\_after',

[$this->\_eventObject => $this, 'source' => $quote]

);

* Event name: sales\_quote\_item\_qty\_set\_after
* File: vendor/magento/Collect/Model/Quote/Item.php

$this->\_eventManager->dispatch('sales\_quote\_item\_qty\_set\_after', ['item' => $this]);

* Event name: sales\_quote\_item\_set\_product
* File: vendor/magento/Collect/Model/Quote/Item.php

$this->\_eventManager->dispatch(

'sales\_quote\_item\_set\_product',

['product' => $product, 'quote\_item' => $this]

);

* Event name: \_import\_data\_before
* File: vendor/magento/Collect/Model/Quote/Payment.php

$this->\_eventManager->dispatch(

$this->\_eventPrefix . '\_import\_data\_before',

[$this->\_eventObject => $this, 'input' => $data]

);

* Event name: \_load\_after
* File: vendor/magento/Collect/Model/ResourceModel/Quote/Address/Collection.php

$this->\_eventManager->dispatch($this->\_eventPrefix . '\_load\_after', [$this->\_eventObject => $this]);

* Event name: prepare\_catalog\_product\_collection\_prices
* File: vendor/magento/Collect/Model/ResourceModel/Quote/Item/Collection.php

$this->\_eventManager->dispatch(

'prepare\_catalog\_product\_collection\_prices',

['collection' => $productCollection, 'store\_id' => $this->getStoreId()]

);

* Event name: sales\_quote\_item\_collection\_products\_after\_load
* File: vendor/magento/Collect/Model/ResourceModel/Quote/Item/Collection.php

$this->\_eventManager->dispatch(

'sales\_quote\_item\_collection\_products\_after\_load',

['collection' => $productCollection]

);

* Event name: controller\_front\_send\_response\_before
* File: vendor/magento/framework/App/Http.php

$this->\_eventManager->dispatch('controller\_front\_send\_response\_before', $eventParams);

* Event name: controller\_action\_layout\_render\_before
* File: vendor/magento/framework/App/View.php

$this->\_eventManager->dispatch('controller\_action\_layout\_render\_before');

* Event name: controller\_action\_layout\_render\_before\_
* File: vendor/magento/framework/App/View.php

$this->\_eventManager->dispatch(

'controller\_action\_layout\_render\_before\_' . $this->\_request->getFullActionName()

);

* Event name: controller\_action\_predispatch
* File: vendor/magento/framework/App/Action/Action.php

$this->\_eventManager->dispatch('controller\_action\_predispatch', $eventParameters);

* Event name: controller\_action\_predispatch\_
* File: vendor/magento/framework/App/Action/Action.php

$this->\_eventManager->dispatch('controller\_action\_predispatch\_' . $request->getRouteName(), $eventParameters);

* Event name: controller\_action\_predispatch\_
* File: vendor/magento/framework/App/Action/Action.php

$this->\_eventManager->dispatch(

'controller\_action\_predispatch\_' . $request->getFullActionName(),

$eventParameters

);

* Event name: controller\_action\_postdispatch\_
* File: vendor/magento/framework/App/Action/Action.php

$this->\_eventManager->dispatch(

'controller\_action\_postdispatch\_' . $request->getFullActionName(),

$eventParameters

);

* Event name: controller\_action\_postdispatch\_
* File: vendor/magento/framework/App/Action/Action.php

$this->\_eventManager->dispatch(

'controller\_action\_postdispatch\_' . $request->getRouteName(),

$eventParameters

);

* Event name: controller\_action\_postdispatch
* File: vendor/magento/framework/App/Action/Action.php

$this->\_eventManager->dispatch('controller\_action\_postdispatch', $eventParameters);

* Event name: controller\_action\_noroute
* File: vendor/magento/framework/Controller/Noroute/Index.php

$this->\_eventManager->dispatch('controller\_action\_noroute', ['action' => $this, 'status' => $status]);

* Event name: currency\_display\_options\_forming
* File: vendor/magento/framework/Locale/Currency.php

$this->\_eventManager->dispatch(

'currency\_display\_options\_forming',

['currency\_options' => $options, 'base\_code' => $currency]

);

* Event name: model\_load\_before
* File: vendor/magento/framework/Model/AbstractModel.php

$this->\_eventManager->dispatch('model\_load\_before', $params);

* Event name: \_load\_before
* File: vendor/magento/framework/Model/AbstractModel.php

$this->\_eventManager->dispatch($this->\_eventPrefix . '\_load\_before', $params);

* Event name: model\_load\_after
* File: vendor/magento/framework/Model/AbstractModel.php

$this->\_eventManager->dispatch('model\_load\_after', ['object' => $this]);

* Event name: \_load\_after
* File: vendor/magento/framework/Model/AbstractModel.php

$this->\_eventManager->dispatch($this->\_eventPrefix . '\_load\_after', $this->\_getEventData());

* Event name: model\_save\_commit\_after
* File: vendor/magento/framework/Model/AbstractModel.php

$this->\_eventManager->dispatch('model\_save\_commit\_after', ['object' => $this]);

* Event name: \_save\_commit\_after
* File: vendor/magento/framework/Model/AbstractModel.php

$this->\_eventManager->dispatch($this->\_eventPrefix . '\_save\_commit\_after', $this->\_getEventData());

* Event name: model\_save\_before
* File: vendor/magento/framework/Model/AbstractModel.php

$this->\_eventManager->dispatch('model\_save\_before', ['object' => $this]);

* Event name: \_save\_before
* File: vendor/magento/framework/Model/AbstractModel.php

$this->\_eventManager->dispatch($this->\_eventPrefix . '\_save\_before', $this->\_getEventData());

* Event name: model\_save\_after
* File: vendor/magento/framework/Model/AbstractModel.php

$this->\_eventManager->dispatch('model\_save\_after', ['object' => $this]);

* Event name: clean\_cache\_by\_tags
* File: vendor/magento/framework/Model/AbstractModel.php

$this->\_eventManager->dispatch('clean\_cache\_by\_tags', ['object' => $this]);

* Event name: \_save\_after
* File: vendor/magento/framework/Model/AbstractModel.php

$this->\_eventManager->dispatch($this->\_eventPrefix . '\_save\_after', $this->\_getEventData());

* Event name: model\_delete\_before
* File: vendor/magento/framework/Model/AbstractModel.php

$this->\_eventManager->dispatch('model\_delete\_before', ['object' => $this]);

* Event name: \_delete\_before
* File: vendor/magento/framework/Model/AbstractModel.php

$this->\_eventManager->dispatch($this->\_eventPrefix . '\_delete\_before', $this->\_getEventData());

* Event name: model\_delete\_after
* File: vendor/magento/framework/Model/AbstractModel.php

$this->\_eventManager->dispatch('model\_delete\_after', ['object' => $this]);

* Event name: clean\_cache\_by\_tags
* File: vendor/magento/framework/Model/AbstractModel.php

$this->\_eventManager->dispatch('clean\_cache\_by\_tags', ['object' => $this]);

* Event name: \_delete\_after
* File: vendor/magento/framework/Model/AbstractModel.php

$this->\_eventManager->dispatch($this->\_eventPrefix . '\_delete\_after', $this->\_getEventData());

* Event name: model\_delete\_commit\_after
* File: vendor/magento/framework/Model/AbstractModel.php

$this->\_eventManager->dispatch('model\_delete\_commit\_after', ['object' => $this]);

* Event name: \_delete\_commit\_after
* File: vendor/magento/framework/Model/AbstractModel.php

$this->\_eventManager->dispatch($this->\_eventPrefix . '\_delete\_commit\_after', $this->\_getEventData());

* Event name: \_clear
* File: vendor/magento/framework/Model/AbstractModel.php

$this->\_eventManager->dispatch($this->\_eventPrefix . '\_clear', $this->\_getEventData());

* Event name: \_clear
* File: vendor/magento/framework/Model/AbstractModel.php

$this->\_eventManager->dispatch($this->\_eventPrefix . '\_clear', $this->\_getEventData());

* Event name: core\_collection\_abstract\_load\_before
* File: vendor/magento/framework/Model/ResourceModel/Db/Collection/AbstractCollection.php

$this->\_eventManager->dispatch('core\_collection\_abstract\_load\_before', ['collection' => $this]);

* Event name: \_load\_before
* File: vendor/magento/framework/Model/ResourceModel/Db/Collection/AbstractCollection.php

$this->\_eventManager->dispatch($this->\_eventPrefix . '\_load\_before', [$this->\_eventObject => $this]);

* Event name: core\_collection\_abstract\_load\_after
* File: vendor/magento/framework/Model/ResourceModel/Db/Collection/AbstractCollection.php

$this->\_eventManager->dispatch('core\_collection\_abstract\_load\_after', ['collection' => $this]);

* Event name: \_load\_after
* File: vendor/magento/framework/Model/ResourceModel/Db/Collection/AbstractCollection.php

$this->\_eventManager->dispatch($this->\_eventPrefix . '\_load\_after', [$this->\_eventObject => $this]);

* Event name: core\_layout\_render\_element
* File: vendor/magento/framework/View/Layout.php

$this->\_eventManager->dispatch(

'core\_layout\_render\_element',

['element\_name' => $name, 'layout' => $this, 'transport' => $this->\_renderingOutput]

);

* Event name: view\_block\_abstract\_to\_html\_before
* File: vendor/magento/framework/View/Element/AbstractBlock.php

$this->\_eventManager->dispatch('view\_block\_abstract\_to\_html\_before', ['block' => $this]);

* Event name: view\_message\_block\_render\_grouped\_html\_after
* File: vendor/magento/framework/View/Element/Messages.php

$this->\_eventManager->dispatch('view\_message\_block\_render\_grouped\_html\_after', $params);

* Event name: adminhtml\_block\_html\_before
* File: vendor/magento/module-backend/Block/Template.php

$this->\_eventManager->dispatch('adminhtml\_block\_html\_before', ['block' => $this]);

* Event name: adminhtml\_store\_edit\_form\_prepare\_form
* File: vendor/magento/module-backend/Block/System/Store/Edit/AbstractForm.php

$this->\_eventManager->dispatch('adminhtml\_store\_edit\_form\_prepare\_form', ['block' => $this]);

* Event name: backend\_block\_widget\_grid\_prepare\_grid\_before
* File: vendor/magento/module-backend/Block/Widget/Grid.php

$this->\_eventManager->dispatch(

'backend\_block\_widget\_grid\_prepare\_grid\_before',

['grid' => $this, 'collection' => $this->getCollection()]

);

* Event name: clean\_catalog\_images\_cache\_after
* File: vendor/magento/module-backend/Controller/Adminhtml/Cache/CleanImages.php

$this->\_eventManager->dispatch('clean\_catalog\_images\_cache\_after');

* Event name: clean\_media\_cache\_after
* File: vendor/magento/module-backend/Controller/Adminhtml/Cache/CleanMedia.php

$this->\_eventManager->dispatch('clean\_media\_cache\_after');

* Event name: clean\_static\_files\_cache\_after
* File: vendor/magento/module-backend/Controller/Adminhtml/Cache/CleanStaticFiles.php

$this->\_eventManager->dispatch('clean\_static\_files\_cache\_after');

* Event name: adminhtml\_cache\_flush\_all
* File: vendor/magento/module-backend/Controller/Adminhtml/Cache/FlushAll.php

$this->\_eventManager->dispatch('adminhtml\_cache\_flush\_all');

* Event name: adminhtml\_cache\_flush\_system
* File: vendor/magento/module-backend/Controller/Adminhtml/Cache/FlushSystem.php

$this->\_eventManager->dispatch('adminhtml\_cache\_flush\_system');

* Event name: theme\_save\_after
* File: vendor/magento/module-backend/Controller/Adminhtml/System/Design/Save.php

$this->\_eventManager->dispatch('theme\_save\_after');

* Event name: store\_delete
* File: vendor/magento/module-backend/Controller/Adminhtml/System/Store/DeleteStorePost.php

$this->\_eventManager->dispatch('store\_delete', ['store' => $model]);

* Event name: store\_group\_save
* File: vendor/magento/module-backend/Controller/Adminhtml/System/Store/Save.php

$this->\_eventManager->dispatch('store\_group\_save', ['group' => $groupModel]);

* Event name: store\_edit
* File: vendor/magento/module-backend/Controller/Adminhtml/System/Store/Save.php

$eventName = 'store\_edit';

$this->\_eventManager->dispatch($eventName, ['store' => $storeModel]);

* Event name: store\_add
* File: vendor/magento/module-backend/Controller/Adminhtml/System/Store/Save.php

$eventName = 'store\_add';

$this->\_eventManager->dispatch($eventName, ['store' => $storeModel]);

* Event name: backend\_auth\_user\_login\_success
* File: vendor/magento/module-backend/Model/Auth.php

$this->\_eventManager->dispatch(

'backend\_auth\_user\_login\_success',

['user' => $this->getCredentialStorage()]

);

* Event name: backend\_auth\_user\_login\_failed
* File: vendor/magento/module-backend/Model/Auth.php

$this->\_eventManager->dispatch(

'backend\_auth\_user\_login\_failed',

['user\_name' => $username, 'exception' => $e]

);

* Event name: catalog\_product\_option\_price\_configuration\_after
* File: vendor/magento/module-bundle/Block/Catalog/Product/View/Type/Bundle.php

$this->\_eventManager->dispatch('catalog\_product\_option\_price\_configuration\_after', ['configObj' => $configObj]);

* Event name: prepare\_catalog\_product\_collection\_prices
* File: vendor/magento/module-bundle/Model/Product/Price.php

$this->\_eventManager->dispatch(

'prepare\_catalog\_product\_collection\_prices',

['collection' => $selections, 'store\_id' => $product->getStoreId()]

);

* Event name: catalog\_product\_get\_final\_price
* File: vendor/magento/module-bundle/Model/Product/Price.php

$this->\_eventManager->dispatch('catalog\_product\_get\_final\_price', ['product' => $product, 'qty' => $qty]);

* Event name: catalog\_product\_prepare\_index\_select
* File: vendor/magento/module-bundle/Model/ResourceModel/Indexer/Price.php

$this->\_eventManager->dispatch(

'catalog\_product\_prepare\_index\_select',

[

'select' => $select,

'entity\_field' => new \Zend\_Db\_Expr('e.entity\_id'),

'website\_field' => new \Zend\_Db\_Expr('cw.website\_id'),

'store\_field' => new \Zend\_Db\_Expr('cs.store\_id')

]

);

* Event name: catalog\_product\_import\_bunch\_delete\_commit\_before
* File: vendor/magento/module-catalog-import-export/Model/Import/Product.php

$this->\_eventManager->dispatch(

'catalog\_product\_import\_bunch\_delete\_commit\_before',

[

'adapter' => $this,

'bunch' => $bunch,

'ids\_to\_delete' => $idsToDelete

]

);

* Event name: catalog\_product\_import\_bunch\_delete\_after
* File: vendor/magento/module-catalog-import-export/Model/Import/Product.php

$this->\_eventManager->dispatch('catalog\_product\_import\_bunch\_delete\_after', ['adapter' => $this, 'bunch' => $bunch]);

* Event name: catalog\_product\_import\_finish\_before
* File: vendor/magento/module-catalog-import-export/Model/Import/Product.php

$this->\_eventManager->dispatch('catalog\_product\_import\_finish\_before', ['adapter' => $this]);

* Event name: catalog\_product\_import\_bunch\_save\_after
* File: vendor/magento/module-catalog-import-export/Model/Import/Product.php

$this->\_eventManager->dispatch(

'catalog\_product\_import\_bunch\_save\_after',

['adapter' => $this, 'bunch' => $bunch]

);

* Event name: catalogrule\_dirty\_notice
* File: vendor/magento/module-catalog-rule/Controller/Adminhtml/Promo/Catalog/Index.php

$this->\_eventManager->dispatch(

'catalogrule\_dirty\_notice',

['dirty\_rules' => $dirtyRules, 'message' => $this->getDirtyRulesNoticeMessage()]

);

* Event name: adminhtml\_controller\_catalogrule\_prepare\_save
* File: vendor/magento/module-catalog-rule/Controller/Adminhtml/Promo/Catalog/Save.php

$this->\_eventManager->dispatch(

'adminhtml\_controller\_catalogrule\_prepare\_save',

['request' => $this->getRequest()]

);

Event name: clean\_cache\_by\_tags

* File: vendor/magento/module-catalog-rule/Model/Indexer/AbstractIndexer.php

$this->\_eventManager->dispatch('clean\_cache\_by\_tags', ['object' => $this]);

* Event name: catalogsearch\_reset\_search\_result
* File: vendor/magento/module-catalog-search/Model/ResourceModel/Fulltext.php

$this->\_eventManager->dispatch('catalogsearch\_reset\_search\_result');

* Event name: shortcut\_buttons\_container
* File: vendor/magento/module-catalog/Block/ShortcutButtons.php

$this->\_eventManager->dispatch(

'shortcut\_buttons\_container',

[

'container' => $this,

'is\_catalog\_product' => $this->\_isCatalogProduct,

'or\_position' => $this->\_orPosition

]

);

* Event name: adminhtml\_catalog\_category\_tree\_is\_moveable
* File: vendor/magento/module-catalog/Block/Adminhtml/Category/Tree.php

$this->\_eventManager->dispatch('adminhtml\_catalog\_category\_tree\_is\_moveable', ['options' => $options]);

* Event name: adminhtml\_catalog\_category\_tree\_can\_add\_root\_category
* File: vendor/magento/module-catalog/Block/Adminhtml/Category/Tree.php

$this->\_eventManager->dispatch(

'adminhtml\_catalog\_category\_tree\_can\_add\_root\_category',

['category' => $this->getCategory(), 'options' => $options, 'store' => $this->getStore()->getId()]

);

* Event name: adminhtml\_catalog\_category\_tree\_can\_add\_sub\_category
* File: vendor/magento/module-catalog/Block/Adminhtml/Category/Tree.php

$this->\_eventManager->dispatch(

'adminhtml\_catalog\_category\_tree\_can\_add\_sub\_category',

['category' => $this->getCategory(), 'options' => $options, 'store' => $this->getStore()->getId()]

);

* Event name: adminhtml\_catalog\_product\_grid\_prepare\_massaction
* File: vendor/magento/module-catalog/Block/Adminhtml/Product/Grid.php

$this->\_eventManager->dispatch('adminhtml\_catalog\_product\_grid\_prepare\_massaction', ['block' => $this]);

* Event name: product\_attribute\_grid\_build
* File: vendor/magento/module-catalog/Block/Adminhtml/Product/Attribute/Grid.php

$this->\_eventManager->dispatch('product\_attribute\_grid\_build', ['grid' => $this]);

* Event name: product\_attribute\_form\_build
* File: vendor/magento/module-catalog/Block/Adminhtml/Product/Attribute/Edit/Tab/Advanced.php

$this->\_eventManager->dispatch('product\_attribute\_form\_build', ['form' => $form]);

* Event name: product\_attribute\_form\_build\_front\_tab
* File: vendor/magento/module-catalog/Block/Adminhtml/Product/Attribute/Edit/Tab/Front.php

$this->\_eventManager->dispatch('product\_attribute\_form\_build\_front\_tab', ['form' => $form]);

* Event name: adminhtml\_catalog\_product\_attribute\_edit\_frontend\_prepare\_form
* File: vendor/magento/module-catalog/Block/Adminhtml/Product/Attribute/Edit/Tab/Front.php

$this->\_eventManager->dispatch(

'adminhtml\_catalog\_product\_attribute\_edit\_frontend\_prepare\_form',

['form' => $form, 'attribute' => $attributeObject]

);

* Event name: adminhtml\_product\_attribute\_types
* File: vendor/magento/module-catalog/Block/Adminhtml/Product/Attribute/Edit/Tab/Main.php

$this->\_eventManager->dispatch('adminhtml\_product\_attribute\_types', ['response' => $response]);

* Event name: product\_attribute\_form\_build\_main\_tab
* File: vendor/magento/module-catalog/Block/Adminhtml/Product/Attribute/Edit/Tab/Main.php

$this->\_eventManager->dispatch('product\_attribute\_form\_build\_main\_tab', ['form' => $form]);

* Event name: adminhtml\_catalog\_product\_edit\_prepare\_form
* File: vendor/magento/module-catalog/Block/Adminhtml/Product/Attribute/NewAttribute/Product/Attributes.php

$this->\_eventManager->dispatch('adminhtml\_catalog\_product\_edit\_prepare\_form', ['form' => $form]);

* Event name: adminhtml\_catalog\_product\_edit\_element\_types
* File: vendor/magento/module-catalog/Block/Adminhtml/Product/Attribute/NewAttribute/Product/Attributes.php

$this->\_eventManager->dispatch('adminhtml\_catalog\_product\_edit\_element\_types', ['response' => $response]);

* Event name: adminhtml\_catalog\_product\_attribute\_set\_main\_html\_before
* File: vendor/magento/module-catalog/Block/Adminhtml/Product/Attribute/Set/Main.php

$this->\_eventManager->dispatch(

'adminhtml\_catalog\_product\_attribute\_set\_main\_html\_before',

['block' => $this]

);

* Event name: adminhtml\_catalog\_product\_attribute\_set\_toolbar\_main\_html\_before
* File: vendor/magento/module-catalog/Block/Adminhtml/Product/Attribute/Set/Toolbar/Main.php

$this->\_eventManager->dispatch(

'adminhtml\_catalog\_product\_attribute\_set\_toolbar\_main\_html\_before',

['block' => $this]

);

* Event name: adminhtml\_catalog\_product\_form\_prepare\_excluded\_field\_list
* File: vendor/magento/module-catalog/Block/Adminhtml/Product/Edit/Action/Attribute/Tab/Attributes.php

$this->\_eventManager->dispatch(

'adminhtml\_catalog\_product\_form\_prepare\_excluded\_field\_list',

['object' => $this]

);

* Event name: adminhtml\_catalog\_product\_edit\_prepare\_form
* File: vendor/magento/module-catalog/Block/Adminhtml/Product/Edit/Tab/Attributes.php

$this->\_eventManager->dispatch(

'adminhtml\_catalog\_product\_edit\_prepare\_form',

['form' => $form, 'layout' => $this->getLayout()]

);

* Event name: adminhtml\_catalog\_product\_edit\_element\_types
* File: vendor/magento/module-catalog/Block/Adminhtml/Product/Edit/Tab/Attributes.php

$this->\_eventManager->dispatch('adminhtml\_catalog\_product\_edit\_element\_types', ['response' => $response]);

* Event name: catalog\_product\_gallery\_prepare\_layout
* File: vendor/magento/module-catalog/Block/Adminhtml/Product/Helper/Form/Gallery/Content.php

$this->\_eventManager->dispatch('catalog\_product\_gallery\_prepare\_layout', ['block' => $this]);

* Event name: catalog\_block\_product\_status\_display
* File: vendor/magento/module-catalog/Block/Product/AbstractProduct.php

$this->\_eventManager->dispatch('catalog\_block\_product\_status\_display', ['status' => $statusInfo]);

* Event name: catalog\_block\_product\_list\_collection
* File: vendor/magento/module-catalog/Block/Product/ListProduct.php

$this->\_eventManager->dispatch(

'catalog\_block\_product\_list\_collection',

['collection' => $this->\_getProductCollection()]

);

* Event name: catalog\_product\_view\_config
* File: vendor/magento/module-catalog/Block/Product/View.php

$this->\_eventManager->dispatch('catalog\_product\_view\_config', ['response\_object' => $responseObject]);

* Event name: catalog\_product\_upsell
* File: vendor/magento/module-catalog/Block/Product/ProductList/Upsell.php

$this->\_eventManager->dispatch(

'catalog\_product\_upsell',

['product' => $product, 'collection' => $this->\_itemCollection, 'limit' => null]

);

* Event name: catalog\_product\_option\_price\_configuration\_after
* File: vendor/magento/module-catalog/Block/Product/View/Options.php

$this->\_eventManager->dispatch('catalog\_product\_option\_price\_configuration\_after', ['configObj' => $configObj]);

* Event name: rss\_catalog\_category\_xml\_callback
* File: vendor/magento/module-catalog/Block/Rss/Category.php

$this->\_eventManager->dispatch('rss\_catalog\_category\_xml\_callback', ['product' => $product]);

* Event name: rss\_catalog\_new\_xml\_callback
* File: vendor/magento/module-catalog/Block/Rss/Product/NewProducts.php

$this->\_eventManager->dispatch('rss\_catalog\_new\_xml\_callback', [

'row' => $item->getData(),

'product' => $item

]);

* Event name: rss\_catalog\_special\_xml\_callback
* File: vendor/magento/module-catalog/Block/Rss/Product/Special.php

$this->\_eventManager->dispatch('rss\_catalog\_special\_xml\_callback', [

'row' => $item->getData(),

'product' => $item

]);

* Event name: category\_prepare\_ajax\_response
* File: vendor/magento/module-catalog/Controller/Adminhtml/Category.php

$this->\_eventManager->dispatch(

'category\_prepare\_ajax\_response',

['response' => $eventResponse, 'controller' => $this]

);

* Event name: catalog\_controller\_category\_delete
* File: vendor/magento/module-catalog/Controller/Adminhtml/Category/Delete.php

$this->\_eventManager->dispatch('catalog\_controller\_category\_delete', ['category' => $category]);

* Event name: catalog\_category\_prepare\_save
* File: vendor/magento/module-catalog/Controller/Adminhtml/Category/Save.php

$this->\_eventManager->dispatch(

'catalog\_category\_prepare\_save',

['category' => $category, 'request' => $this->getRequest()]

);

* Event name: catalog\_product\_edit\_action
* File: vendor/magento/module-catalog/Controller/Adminhtml/Product/Edit.php

$this->\_eventManager->dispatch('catalog\_product\_edit\_action', ['product' => $product]);

* Event name: catalog\_product\_new\_action
* File: vendor/magento/module-catalog/Controller/Adminhtml/Product/NewAction.php

$this->\_eventManager->dispatch('catalog\_product\_new\_action', ['product' => $product]);

* Event name: controller\_action\_catalog\_product\_save\_entity\_after
* File: vendor/magento/module-catalog/Controller/Adminhtml/Product/Save.php

$this->\_eventManager->dispatch(

'controller\_action\_catalog\_product\_save\_entity\_after',

['controller' => $this, 'product' => $product]

);

* Event name: catalog\_product\_to\_website\_change
* File: vendor/magento/module-catalog/Controller/Adminhtml/Product/Action/Attribute/Save.php

$this->\_eventManager->dispatch('catalog\_product\_to\_website\_change', ['products' => $productIds]);

* Event name: catalog\_controller\_category\_init\_after
* File: vendor/magento/module-catalog/Controller/Category/View.php

$this->\_eventManager->dispatch(

'catalog\_controller\_category\_init\_after',

['category' => $category, 'controller\_action' => $this]

);

* Event name: catalog\_product\_compare\_add\_product
* File: vendor/magento/module-catalog/Controller/Product/Compare/Add.php

$this->\_eventManager->dispatch('catalog\_product\_compare\_add\_product', ['product' => $product]);

* Event name: catalog\_product\_compare\_remove\_product
* File: vendor/magento/module-catalog/Controller/Product/Compare/Remove.php

$this->\_eventManager->dispatch(

'catalog\_product\_compare\_remove\_product',

['product' => $item]

);

* Event name: catalog\_controller\_product\_init\_before
* File: vendor/magento/module-catalog/Helper/Product.php

$this->\_eventManager->dispatch(

'catalog\_controller\_product\_init\_before',

['controller\_action' => $controller, 'params' => $params]

);

* Event name: catalog\_controller\_product\_init\_after
* File: vendor/magento/module-catalog/Helper/Product.php

$this->\_eventManager->dispatch(

'catalog\_controller\_product\_init\_after',

['product' => $product, 'controller\_action' => $controller]

);

* Event name: catalog\_controller\_product\_view
* File: vendor/magento/module-catalog/Helper/Product/View.php

$this->\_eventManager->dispatch('catalog\_controller\_product\_view', ['product' => $product]);

* Event name: \_move\_before
* File: vendor/magento/module-catalog/Model/Category.php

$this->\_eventManager->dispatch($this->\_eventPrefix . '\_move\_before', $eventParams);

* Event name: \_move\_after
* File: vendor/magento/module-catalog/Model/Category.php

$this->\_eventManager->dispatch($this->\_eventPrefix . '\_move\_after', $eventParams);

* Event name: category\_move
* File: vendor/magento/module-catalog/Model/Category.php

$this->\_eventManager->dispatch($this->\_eventPrefix . 'category\_move', $eventParams);

* Event name: clean\_cache\_by\_tags
* File: vendor/magento/module-catalog/Model/Category.php

$this->\_eventManager->dispatch('clean\_cache\_by\_tags', ['object' => $this]);

* Event name: \_validate\_before
* File: vendor/magento/module-catalog/Model/Product.php

$this->\_eventManager->dispatch($this->\_eventPrefix . '\_validate\_before', $this->\_getEventData());

* Event name: \_validate\_after
* File: vendor/magento/module-catalog/Model/Product.php

$this->\_eventManager->dispatch($this->\_eventPrefix . '\_validate\_after', $this->\_getEventData());

* Event name: catalog\_product\_is\_salable\_before
* File: vendor/magento/module-catalog/Model/Product.php

$this->\_eventManager->dispatch('catalog\_product\_is\_salable\_before', ['product' => $this]);

* Event name: catalog\_product\_is\_salable\_after
* File: vendor/magento/module-catalog/Model/Product.php

$this->\_eventManager->dispatch(

'catalog\_product\_is\_salable\_after',

['product' => $this, 'salable' => $object]

);

* Event name: catalog\_product\_attribute\_update\_before
* File: vendor/magento/module-catalog/Model/Product/Action.php

$this->\_eventManager->dispatch(

'catalog\_product\_attribute\_update\_before',

['attributes\_data' => &$attrData, 'product\_ids' => &$productIds, 'store\_id' => &$storeId]

);

* Event name: adminhtml\_product\_attribute\_types
* File: vendor/magento/module-catalog/Model/Product/Attribute/Source/Inputtype.php

$this->\_eventManager->dispatch('adminhtml\_product\_attribute\_types', ['response' => $response]);

* Event name: catalog\_product\_type\_prepare\_%s\_options
* File: vendor/magento/module-catalog/Model/Product/Type/AbstractType.php

$eventName = sprintf('catalog\_product\_type\_prepare\_%s\_options', $processMode);

$this->\_eventManager->dispatch(

$eventName,

['transport' => $transport, 'buy\_request' => $buyRequest, 'product' => $product]

);

* Event name: catalog\_product\_get\_final\_price
* File: vendor/magento/module-catalog/Model/Product/Type/Price.php

$this->\_eventManager->dispatch('catalog\_product\_get\_final\_price', ['product' => $product, 'qty' => $qty]);

* Event name: catalog\_category\_change\_products
* File: vendor/magento/module-catalog/Model/ResourceModel/Category.php

$this->\_eventManager->dispatch(

'catalog\_category\_change\_products',

['category' => $category, 'product\_ids' => $productIds]

);

* Event name: catalog\_category\_delete\_after\_done
* File: vendor/magento/module-catalog/Model/ResourceModel/Category.php

$this->\_eventManager->dispatch(

'catalog\_category\_delete\_after\_done',

['product' => $object]

);

* Event name: \_load\_before
* File: vendor/magento/module-catalog/Model/ResourceModel/Category/Collection.php

$this->\_eventManager->dispatch($this->\_eventPrefix . '\_load\_before', [$this->\_eventObject => $this]);

* Event name: \_load\_after
* File: vendor/magento/module-catalog/Model/ResourceModel/Category/Collection.php

$this->\_eventManager->dispatch($this->\_eventPrefix . '\_load\_after', [$this->\_eventObject => $this]);

* Event name: \_add\_is\_active\_filter
* File: vendor/magento/module-catalog/Model/ResourceModel/Category/Collection.php

$this->\_eventManager->dispatch(

$this->\_eventPrefix . '\_add\_is\_active\_filter',

[$this->\_eventObject => $this]

);

* Event name: catalog\_category\_tree\_init\_inactive\_category\_ids
* File: vendor/magento/module-catalog/Model/ResourceModel/Category/Flat.php

$this->\_eventManager->dispatch('catalog\_category\_tree\_init\_inactive\_category\_ids', ['tree' => $this]);

* Event name: catalog\_category\_flat\_loadnodes\_before
* File: vendor/magento/module-catalog/Model/ResourceModel/Category/Flat.php

$this->\_eventManager->dispatch('catalog\_category\_flat\_loadnodes\_before', ['select' => $select]);

* Event name: catalog\_category\_tree\_init\_inactive\_category\_ids
* File: vendor/magento/module-catalog/Model/ResourceModel/Category/Tree.php

$this->\_eventManager->dispatch('catalog\_category\_tree\_init\_inactive\_category\_ids', ['tree' => $this]);

* Event name: \_load\_before
* File: vendor/magento/module-catalog/Model/ResourceModel/Category/Flat/Collection.php

$this->\_eventManager->dispatch($this->\_eventPrefix . '\_load\_before', [$this->\_eventObject => $this]);

* Event name: \_load\_after
* File: vendor/magento/module-catalog/Model/ResourceModel/Category/Flat/Collection.php

$this->\_eventManager->dispatch($this->\_eventPrefix . '\_load\_after', [$this->\_eventObject => $this]);

* Event name: \_add\_is\_active\_filter
* File: vendor/magento/module-catalog/Model/ResourceModel/Category/Flat/Collection.php

$this->\_eventManager->dispatch(

$this->\_eventPrefix . '\_add\_is\_active\_filter',

[$this->\_eventObject => $this]

);

* Event name: catalog\_prepare\_price\_select
* File: vendor/magento/module-catalog/Model/ResourceModel/Product/Collection.php

$this->\_eventManager->dispatch('catalog\_prepare\_price\_select', $eventArgs);

* Event name: catalog\_product\_collection\_load\_after
* File: vendor/magento/module-catalog/Model/ResourceModel/Product/Collection.php

$this->\_eventManager->dispatch('catalog\_product\_collection\_load\_after', ['collection' => $this]);

* Event name: catalog\_product\_collection\_before\_add\_count\_to\_categories
* File: vendor/magento/module-catalog/Model/ResourceModel/Product/Collection.php

$this->\_eventManager->dispatch(

'catalog\_product\_collection\_before\_add\_count\_to\_categories',

['collection' => $this]

);

* Event name: catalog\_product\_collection\_apply\_limitations\_after
* File: vendor/magento/module-catalog/Model/ResourceModel/Product/Collection.php

$this->\_eventManager->dispatch(

'catalog\_product\_collection\_apply\_limitations\_after',

['collection' => $this]

);

* Event name: catalog\_product\_compare\_item\_collection\_clear
* File: vendor/magento/module-catalog/Model/ResourceModel/Product/Compare/Item/Collection.php

$this->\_eventManager->dispatch('catalog\_product\_compare\_item\_collection\_clear');

* Event name: prepare\_catalog\_product\_index\_select
* File: vendor/magento/module-catalog/Model/ResourceModel/Product/Indexer/Eav/AbstractEav.php

$this->\_eventManager->dispatch(

'prepare\_catalog\_product\_index\_select',

[

'select' => $select,

'entity\_field' => new \Zend\_Db\_Expr('l.parent\_id'),

'website\_field' => new \Zend\_Db\_Expr('cs.website\_id'),

'store\_field' => new \Zend\_Db\_Expr('cs.store\_id')

]

);

* Event name: prepare\_catalog\_product\_index\_select
* File: vendor/magento/module-catalog/Model/ResourceModel/Product/Indexer/Eav/Decimal.php

$this->\_eventManager->dispatch(

'prepare\_catalog\_product\_index\_select',

[

'select' => $select,

'entity\_field' => new \Zend\_Db\_Expr('cpe.entity\_id'),

'website\_field' => new \Zend\_Db\_Expr('cs.website\_id'),

'store\_field' => new \Zend\_Db\_Expr('cs.store\_id')

]

);

* Event name: prepare\_catalog\_product\_index\_select
* File: vendor/magento/module-catalog/Model/ResourceModel/Product/Indexer/Eav/Source.php

$this->\_eventManager->dispatch(

'prepare\_catalog\_product\_index\_select',

[

'select' => $select,

'entity\_field' => new \Zend\_Db\_Expr('pid.entity\_id'),

'website\_field' => new \Zend\_Db\_Expr('pid.website\_id'),

'store\_field' => new \Zend\_Db\_Expr('pid.store\_id')

]

);

* Event name: prepare\_catalog\_product\_index\_select
* File: vendor/magento/module-catalog/Model/ResourceModel/Product/Indexer/Eav/Source.php

$this->\_eventManager->dispatch(

'prepare\_catalog\_product\_index\_select',

[ 'select' => $select,

'entity\_field' => new \Zend\_Db\_Expr('cpe.entity\_id'),

'website\_field' => new \Zend\_Db\_Expr('cs.website\_id'),

'store\_field' => new \Zend\_Db\_Expr('cs.store\_id')

]

);

* Event name: prepare\_catalog\_product\_index\_select
* File: vendor/magento/module-catalog/Model/ResourceModel/Product/Indexer/Price/DefaultPrice.php

$this->\_eventManager->dispatch(

'prepare\_catalog\_product\_index\_select',

[

'select' => $select,

'entity\_field' => new \Zend\_Db\_Expr('e.entity\_id'),

'website\_field' => new \Zend\_Db\_Expr('cw.website\_id'),

'store\_field' => new \Zend\_Db\_Expr('cs.store\_id')

]

);

* Event name: shortcut\_buttons\_container
* File: vendor/magento/module-checkout/Block/QuoteShortcutButtons.php

$this->\_eventManager->dispatch(

'shortcut\_buttons\_container',

[

'container' => $this,

'is\_catalog\_product' => $this->\_isCatalogProduct,

'or\_position' => $this->\_orPosition,

'checkout\_session' => $this->\_checkoutSession

]

);

* Event name: checkout\_cart\_add\_product\_complete
* File: vendor/magento/module-checkout/Controller/Cart/Add.php

$this->\_eventManager->dispatch(

'checkout\_cart\_add\_product\_complete',

['product' => $product, 'request' => $this->getRequest(), 'response' => $this->getResponse()]

);

* Event name: checkout\_cart\_update\_item\_complete
* File: vendor/magento/module-checkout/Controller/Cart/UpdateItemOptions.php

$this->\_eventManager->dispatch(

'checkout\_cart\_update\_item\_complete',

['item' => $item, 'request' => $this->getRequest(), 'response' => $this->getResponse()]

);

* Event name: checkout\_controller\_onepage\_saveOrder
* File: vendor/magento/module-checkout/Controller/Onepage/SaveOrder.php

$this->\_eventManager->dispatch(

'checkout\_controller\_onepage\_saveOrder',

[

'result' => $result,

'action' => $this

]

);

* Event name: checkout\_onepage\_controller\_success\_action
* File: vendor/magento/module-checkout/Controller/Onepage/Success.php

$this->\_eventManager->dispatch(

'checkout\_onepage\_controller\_success\_action',

['order\_ids' => [$session->getLastOrderId()]]

);

* Event name: checkout\_allow\_guest
* File: vendor/magento/module-checkout/Helper/Data.php

$this->\_eventManager->dispatch(

'checkout\_allow\_guest',

['quote' => $quote, 'store' => $store, 'result' => $result]

);

* Event name: checkout\_cart\_product\_add\_after
* File: vendor/magento/module-checkout/Model/Cart.php

$this->\_eventManager->dispatch(

'checkout\_cart\_product\_add\_after',

['quote\_item' => $result, 'product' => $product]

);

* Event name: checkout\_cart\_update\_items\_before
* File: vendor/magento/module-checkout/Model/Cart.php

$this->\_eventManager->dispatch(

'checkout\_cart\_update\_items\_before',

['cart' => $this, 'info' => $infoDataObject]

);

* Event name: checkout\_cart\_update\_items\_after
* File: vendor/magento/module-checkout/Model/Cart.php

$this->\_eventManager->dispatch(

'checkout\_cart\_update\_items\_after',

['cart' => $this, 'info' => $infoDataObject]

);

* Event name: checkout\_cart\_save\_before
* File: vendor/magento/module-checkout/Model/Cart.php

$this->\_eventManager->dispatch('checkout\_cart\_save\_before', ['cart' => $this]);

* Event name: checkout\_cart\_save\_after
* File: vendor/magento/module-checkout/Model/Cart.php

$this->\_eventManager->dispatch('checkout\_cart\_save\_after', ['cart' => $this]);

* Event name: checkout\_cart\_product\_update\_after
* File: vendor/magento/module-checkout/Model/Cart.php

$this->\_eventManager->dispatch(

'checkout\_cart\_product\_update\_after',

['quote\_item' => $result, 'product' => $product]

);

* Event name: custom\_quote\_process
* File: vendor/magento/module-checkout/Model/Session.php

$this->\_eventManager->dispatch('custom\_quote\_process', ['checkout\_session' => $this]);

* Event name: checkout\_quote\_init
* File: vendor/magento/module-checkout/Model/Session.php

$this->\_eventManager->dispatch('checkout\_quote\_init', ['quote' => $quote]);

* Event name: load\_customer\_quote\_before
* File: vendor/magento/module-checkout/Model/Session.php

$this->\_eventManager->dispatch('load\_customer\_quote\_before', ['checkout\_session' => $this]);

* Event name: checkout\_quote\_destroy
* File: vendor/magento/module-checkout/Model/Session.php

$this->\_eventManager->dispatch('checkout\_quote\_destroy', ['quote' => $this->getQuote()]);

* Event name: restore\_quote
* File: vendor/magento/module-checkout/Model/Session.php

$this->\_eventManager->dispatch('restore\_quote', ['order' => $order, 'quote' => $quote]);

* Event name: checkout\_type\_onepage\_save\_order\_after
* File: vendor/magento/module-checkout/Model/Type/Onepage.php

$this->\_eventManager->dispatch(

'checkout\_type\_onepage\_save\_order\_after',

['order' => $order, 'quote' => $this->getQuote()]

);

* Event name: checkout\_submit\_all\_after
* File: vendor/magento/module-checkout/Model/Type/Onepage.php

$this->\_eventManager->dispatch(

'checkout\_submit\_all\_after',

[

'order' => $order,

'quote' => $this->getQuote()

]

);

* Event name: cms\_controller\_router\_match\_before
* File: vendor/magento/module-cms/Controller/Router.php

$this->\_eventManager->dispatch(

'cms\_controller\_router\_match\_before',

['router' => $this, 'condition' => $condition]

);

* Event name: adminhtml\_cmspage\_on\_delete
* File: vendor/magento/module-cms/Controller/Adminhtml/Page/Delete.php

$this->\_eventManager->dispatch(

'adminhtml\_cmspage\_on\_delete',

['title' => $title, 'status' => 'success']

);

* Event name: adminhtml\_cmspage\_on\_delete
* File: vendor/magento/module-cms/Controller/Adminhtml/Page/Delete.php

$this->\_eventManager->dispatch(

'adminhtml\_cmspage\_on\_delete',

['title' => $title, 'status' => 'fail']

);

* Event name: cms\_page\_prepare\_save
* File: vendor/magento/module-cms/Controller/Adminhtml/Page/Save.php

$this->\_eventManager->dispatch(

'cms\_page\_prepare\_save',

['page' => $model, 'request' => $this->getRequest()]

);

* Event name: cms\_page\_render
* File: vendor/magento/module-cms/Helper/Page.php

$this->\_eventManager->dispatch(

'cms\_page\_render',

['page' => $this->\_page, 'controller\_action' => $action]

);

* Event name: cms\_wysiwyg\_images\_static\_urls\_allowed
* File: vendor/magento/module-cms/Helper/Wysiwyg/Images.php

$this->\_eventManager->dispatch(

'cms\_wysiwyg\_images\_static\_urls\_allowed',

['result' => $checkResult, 'store\_id' => $this->\_storeId]

);

* Event name: adminhtml\_system\_config\_advanced\_disableoutput\_render\_before
* File: vendor/magento/module-config/Block/System/Config/Form/Fieldset/Modules/DisableOutput.php

$this->\_eventManager->dispatch(

'adminhtml\_system\_config\_advanced\_disableoutput\_render\_before',

['modules' => $dispatchResult]

);

* Event name: admin\_system\_config\_changed\_section\_
* File: vendor/magento/module-config/Model/Config.php

$this->\_eventManager->dispatch(

"admin\_system\_config\_changed\_section\_{$this->getSection()}",

['website' => $this->getWebsite(), 'store' => $this->getStore()]

);

* Event name: controller\_action\_nocookies
* File: vendor/magento/module-cookie/Controller/Index/NoCookies.php

$this->\_eventManager->dispatch(

'controller\_action\_nocookies',

['action' => $this, 'redirect' => $redirect]

);

* Event name: admin\_system\_config\_changed\_section\_currency\_before\_reinit
* File: vendor/magento/module-currency-symbol/Model/System/Currencysymbol.php

$this->\_eventManager->dispatch(

'admin\_system\_config\_changed\_section\_currency\_before\_reinit',

['website' => $this->\_websiteId, 'store' => $this->\_storeId]

);

* Event name: adminhtml\_block\_html\_before
* File: vendor/magento/module-customer/Block/Adminhtml/Edit/Tab/Carts.php

$this->\_eventManager->dispatch('adminhtml\_block\_html\_before', ['block' => $this]);

* Event name: customer\_register\_success
* File: vendor/magento/module-customer/Controller/Account/CreatePost.php

$this->\_eventManager->dispatch(

'customer\_register\_success',

['account\_controller' => $this, 'customer' => $customer]

);

* Event name: customer\_account\_edited
* File: vendor/magento/module-customer/Controller/Account/EditPost.php

$this->\_eventManager->dispatch(

'customer\_account\_edited',

['email' => $customerCandidateDataObject->getEmail()]

);

* Event name: adminhtml\_customer\_prepare\_save
* File: vendor/magento/module-customer/Controller/Adminhtml/Index/Save.php

$this->\_eventManager->dispatch(

'adminhtml\_customer\_prepare\_save',

['customer' => $customer, 'request' => $this->getRequest()]

);

* Event name: adminhtml\_customer\_save\_after
* File: vendor/magento/module-customer/Controller/Adminhtml/Index/Save.php

$this->\_eventManager->dispatch(

'adminhtml\_customer\_save\_after',

['customer' => $customer, 'request' => $this->getRequest()]

);

* Event name: customer\_customer\_authenticated
* File: vendor/magento/module-customer/Model/Customer.php

$this->\_eventManager->dispatch(

'customer\_customer\_authenticated',

['model' => $this, 'password' => $password]

);

* Event name: customer\_session\_init
* File: vendor/magento/module-customer/Model/Session.php

$this->\_eventManager->dispatch('customer\_session\_init', ['customer\_session' => $this]);

* Event name: customer\_login
* File: vendor/magento/module-customer/Model/Session.php

$this->\_eventManager->dispatch('customer\_login', ['customer' => $customer]);

* Event name: customer\_data\_object\_login
* File: vendor/magento/module-customer/Model/Session.php

$this->\_eventManager->dispatch('customer\_data\_object\_login', ['customer' => $this->getCustomerDataObject()]);

* Event name: customer\_logout
* File: vendor/magento/module-customer/Model/Session.php

$this->\_eventManager->dispatch('customer\_logout', ['customer' => $this->getCustomer()]);

* Event name: visitor\_init
* File: vendor/magento/module-customer/Model/Visitor.php

$this->\_eventManager->dispatch('visitor\_init', ['visitor' => $this]);

* Event name: visitor\_activity\_save
* File: vendor/magento/module-customer/Model/Visitor.php

$this->\_eventManager->dispatch('visitor\_activity\_save', ['visitor' => $this]);

* Event name: customer\_address\_format
* File: vendor/magento/module-customer/Model/Address/AbstractAddress.php

$this->\_eventManager->dispatch('customer\_address\_format', ['type' => $formatType, 'address' => $this]);

* Event name: adminhtml\_block\_eav\_attribute\_edit\_form\_init
* File: vendor/magento/module-eav/Block/Adminhtml/Attribute/Edit/Main/AbstractMain.php

$this->\_eventManager->dispatch(

'adminhtml\_block\_eav\_attribute\_edit\_form\_init',

['form' => $this->getForm()]

);

* Event name: eav\_collection\_abstract\_load\_before
* File: vendor/magento/module-eav/Model/Entity/Collection/AbstractCollection.php

$this->\_eventManager->dispatch('eav\_collection\_abstract\_load\_before', ['collection' => $this]);

* Event name: gift\_options\_prepare\_items
* File: vendor/magento/module-gift-message/Block/Message/Inline.php

$this->\_eventManager->dispatch('gift\_options\_prepare\_items', ['items' => $entityItems]);

* Event name: catalog\_product\_prepare\_index\_select
* File: vendor/magento/module-grouped-product/Model/ResourceModel/Product/Indexer/Price/Grouped.php

$this->\_eventManager->dispatch(

'catalog\_product\_prepare\_index\_select',

[

'select' => $select,

'entity\_field' => new \Zend\_Db\_Expr('e.entity\_id'),

'website\_field' => new \Zend\_Db\_Expr('cw.website\_id'),

'store\_field' => new \Zend\_Db\_Expr('cs.store\_id')

]

);

* Event name: checkout\_controller\_multishipping\_shipping\_post
* File: vendor/magento/module-multishipping/Controller/Checkout/ShippingPost.php

$this->\_eventManager->dispatch(

'checkout\_controller\_multishipping\_shipping\_post',

['request' => $this->getRequest(), 'quote' => $this->\_getCheckout()->getQuote()]

);

* Event name: multishipping\_checkout\_controller\_success\_action
* File: vendor/magento/module-multishipping/Controller/Checkout/Success.php

$this->\_eventManager->dispatch('multishipping\_checkout\_controller\_success\_action', ['order\_ids' => $ids]);

* Event name: checkout\_type\_multishipping\_set\_shipping\_items
* File: vendor/magento/module-multishipping/Model/Checkout/Type/Multishipping.php

$this->\_eventManager->dispatch('checkout\_type\_multishipping\_set\_shipping\_items', ['quote' => $quote]);

* Event name: checkout\_type\_multishipping\_create\_orders\_single
* File: vendor/magento/module-multishipping/Model/Checkout/Type/Multishipping.php

$this->\_eventManager->dispatch(

'checkout\_type\_multishipping\_create\_orders\_single',

['order' => $order, 'address' => $address, 'quote' => $this->getQuote()]

);

* Event name: checkout\_submit\_all\_after
* File: vendor/magento/module-multishipping/Model/Checkout/Type/Multishipping.php

$this->\_eventManager->dispatch(

'checkout\_submit\_all\_after',

['orders' => $orders, 'quote' => $this->getQuote()]

);

* Event name: checkout\_multishipping\_refund\_all
* File: vendor/magento/module-multishipping/Model/Checkout/Type/Multishipping.php

$this->\_eventManager->dispatch('checkout\_multishipping\_refund\_all', ['orders' => $orders]);

* Event name: payment\_form\_block\_to\_html\_before
* File: vendor/magento/module-payment/Block/Form/Cc.php

$this->\_eventManager->dispatch('payment\_form\_block\_to\_html\_before', ['block' => $this]);

* Event name: payment\_cart\_collect\_items\_and\_amounts
* File: vendor/magento/module-payment/Model/Cart.php

$this->\_eventManager->dispatch('payment\_cart\_collect\_items\_and\_amounts', ['cart' => $this]);

* Event name: payment\_method\_assign\_data\_
* File: vendor/magento/module-payment/Model/Method/AbstractMethod.php

$this->\_eventManager->dispatch(

'payment\_method\_assign\_data\_' . $this->getCode(),

[

AbstractDataAssignObserver::METHOD\_CODE => $this,

AbstractDataAssignObserver::MODEL\_CODE => $this->getInfoInstance(),

AbstractDataAssignObserver::DATA\_CODE => $data

]

);

* Event name: payment\_method\_assign\_data
* File: vendor/magento/module-payment/Model/Method/AbstractMethod.php

$this->\_eventManager->dispatch(

'payment\_method\_assign\_data',

[

AbstractDataAssignObserver::METHOD\_CODE => $this,

AbstractDataAssignObserver::MODEL\_CODE => $this->getInfoInstance(),

AbstractDataAssignObserver::DATA\_CODE => $data

]

);

* Event name: payment\_method\_is\_active
* File: vendor/magento/module-payment/Model/Method/AbstractMethod.php

$this->\_eventManager->dispatch(

'payment\_method\_is\_active',

[

'result' => $checkResult,

'method\_instance' => $this,

'quote' => $quote

]

);

* Event name: paypal\_express\_place\_order\_success
* File: vendor/magento/module-paypal/Controller/Express/AbstractExpress/PlaceOrder.php

$this->\_eventManager->dispatch(

'paypal\_express\_place\_order\_success',

[

'order' => $order,

'quote' => $this->\_getQuote()

]

);

* Event name: persistent\_session\_expired
* File: vendor/magento/module-persistent/Controller/Index/UnsetCookie.php

$this->\_eventManager->dispatch('persistent\_session\_expired');

* Event name: persistent\_session\_expired
* File: vendor/magento/module-persistent/Observer/CheckExpirePersistentQuoteObserver.php

$this->\_eventManager->dispatch('persistent\_session\_expired');

* Event name: sales\_quote\_remove\_item
* File: vendor/magento/module-quote/Model/Quote.php

$this->\_eventManager->dispatch('sales\_quote\_remove\_item', ['quote\_item' => $item]);

* Event name: sales\_quote\_add\_item
* File: vendor/magento/module-quote/Model/Quote.php

$this->\_eventManager->dispatch('sales\_quote\_add\_item', ['quote\_item' => $item]);

* Event name: sales\_quote\_product\_add\_after
* File: vendor/magento/module-quote/Model/Quote.php

$this->\_eventManager->dispatch('sales\_quote\_product\_add\_after', ['items' => $items]);

* Event name: \_merge\_before
* File: vendor/magento/module-quote/Model/Quote.php

$this->\_eventManager->dispatch(

$this->\_eventPrefix . '\_merge\_before',

[$this->\_eventObject => $this, 'source' => $quote]

);

* Event name: \_merge\_after
* File: vendor/magento/module-quote/Model/Quote.php

$this->\_eventManager->dispatch(

$this->\_eventPrefix . '\_merge\_after',

[$this->\_eventObject => $this, 'source' => $quote]

);

* Event name: sales\_quote\_item\_qty\_set\_after
* File: vendor/magento/module-quote/Model/Quote/Item.php

$this->\_eventManager->dispatch('sales\_quote\_item\_qty\_set\_after', ['item' => $this]);

* Event name: sales\_quote\_item\_set\_product
* File: vendor/magento/module-quote/Model/Quote/Item.php

$this->\_eventManager->dispatch(

'sales\_quote\_item\_set\_product',

['product' => $product, 'quote\_item' => $this]

);

* Event name: \_import\_data\_before
* File: vendor/magento/module-quote/Model/Quote/Payment.php

$this->\_eventManager->dispatch(

$this->\_eventPrefix . '\_import\_data\_before',

[$this->\_eventObject => $this, 'input' => $data]

);

* Event name: \_load\_after
* File: vendor/magento/module-quote/Model/ResourceModel/Quote/Address/Collection.php

$this->\_eventManager->dispatch($this->\_eventPrefix . '\_load\_after', [$this->\_eventObject => $this]);

* Event name: prepare\_catalog\_product\_collection\_prices
* File: vendor/magento/module-quote/Model/ResourceModel/Quote/Item/Collection.php

$this->\_eventManager->dispatch(

'prepare\_catalog\_product\_collection\_prices',

['collection' => $productCollection, 'store\_id' => $this->getStoreId()]

);

* Event name: sales\_quote\_item\_collection\_products\_after\_load
* File: vendor/magento/module-quote/Model/ResourceModel/Quote/Item/Collection.php

$this->\_eventManager->dispatch(

'sales\_quote\_item\_collection\_products\_after\_load',

['collection' => $productCollection]

);

* Event name: adminhtml\_widget\_grid\_filter\_collection
* File: vendor/magento/module-reports/Block/Adminhtml/Grid.php

$this->\_eventManager->dispatch(

'adminhtml\_widget\_grid\_filter\_collection',

['collection' => $this->getCollection(), 'filter\_values' => $this->\_filterValues]

);

* Event name: sales\_prepare\_amount\_expression
* File: vendor/magento/module-reports/Model/ResourceModel/Order/Collection.php

$this->\_eventManager->dispatch(

'sales\_prepare\_amount\_expression',

['collection' => $this, 'expression\_object' => $expressionTransferObject]

);

* Event name: review\_controller\_product\_init\_before
* File: vendor/magento/module-review/Controller/Product.php

$this->\_eventManager->dispatch('review\_controller\_product\_init\_before', ['controller\_action' => $this]);

* Event name: review\_controller\_product\_init
* File: vendor/magento/module-review/Controller/Product.php

$this->\_eventManager->dispatch('review\_controller\_product\_init', ['product' => $product]);

* Event name: review\_controller\_product\_init\_after
* File: vendor/magento/module-review/Controller/Product.php

$this->\_eventManager->dispatch(

'review\_controller\_product\_init\_after',

['product' => $product, 'controller\_action' => $this]

);

* Event name: rating\_rating\_collection\_load\_before
* File: vendor/magento/module-review/Model/ResourceModel/Rating/Collection.php

$this->\_eventManager->dispatch('rating\_rating\_collection\_load\_before', ['collection' => $this]);

* Event name: review\_review\_collection\_load\_before
* File: vendor/magento/module-review/Model/ResourceModel/Review/Collection.php

$this->\_eventManager->dispatch('review\_review\_collection\_load\_before', ['collection' => $this]);

* Event name: adminhtml\_block\_salesrule\_actions\_prepareform
* File: vendor/magento/module-sales-rule/Block/Adminhtml/Promo/Quote/Edit/Tab/Actions.php

$this->\_eventManager->dispatch('adminhtml\_block\_salesrule\_actions\_prepareform', ['form' => $form]);

* Event name: adminhtml\_promo\_quote\_edit\_tab\_coupons\_form\_prepare\_form
* File: vendor/magento/module-sales-rule/Block/Adminhtml/Promo/Quote/Edit/Tab/Coupons/Form.php

$this->\_eventManager->dispatch(

'adminhtml\_promo\_quote\_edit\_tab\_coupons\_form\_prepare\_form',

['form' => $form]

);

* Event name: adminhtml\_block\_promo\_widget\_chooser\_prepare\_collection
* File: vendor/magento/module-sales-rule/Block/Adminhtml/Promo/Widget/Chooser.php

$this->\_eventManager->dispatch(

'adminhtml\_block\_promo\_widget\_chooser\_prepare\_collection',

['collection' => $collection]

);

* Event name: adminhtml\_controller\_salesrule\_prepare\_save
* File: vendor/magento/module-sales-rule/Controller/Adminhtml/Promo/Quote/Save.php

$this->\_eventManager->dispatch(

'adminhtml\_controller\_salesrule\_prepare\_save',

['request' => $this->getRequest()]

);

* Event name: salesrule\_rule\_get\_coupon\_types
* File: vendor/magento/module-sales-rule/Model/Rule.php

$this->\_eventManager->dispatch('salesrule\_rule\_get\_coupon\_types', ['transport' => $transport]);

* Event name: salesrule\_validator\_process
* File: vendor/magento/module-sales-rule/Model/RulesApplier.php

$this->\_eventManager->dispatch(

'salesrule\_validator\_process',

[

'rule' => $rule,

'item' => $item,

'address' => $address,

'quote' => $quote,

'qty' => $qty,

'result' => $discountData

]

);

* Event name: salesrule\_rule\_condition\_combine
* File: vendor/magento/module-sales-rule/Model/Rule/Condition/Combine.php

$this->\_eventManager->dispatch('salesrule\_rule\_condition\_combine', ['additional' => $additional]);

* Event name: adminhtml\_customer\_orders\_add\_action\_renderer
* File: vendor/magento/module-sales/Block/Adminhtml/Reorder/Renderer/Action.php

$this->\_eventManager->dispatch(

'adminhtml\_customer\_orders\_add\_action\_renderer',

['renderer' => $this, 'row' => $row]

);

* Event name: admin\_sales\_order\_address\_update
* File: vendor/magento/module-sales/Controller/Adminhtml/Order/AddressSave.php

$this->\_eventManager->dispatch(

'admin\_sales\_order\_address\_update',

[

'order\_id' => $address->getParentId()

]

);

* Event name: adminhtml\_sales\_order\_create\_process\_data\_before
* File: vendor/magento/module-sales/Controller/Adminhtml/Order/Create.php

$this->\_eventManager->dispatch('adminhtml\_sales\_order\_create\_process\_data\_before', $eventData);

* Event name: adminhtml\_sales\_order\_create\_process\_data
* File: vendor/magento/module-sales/Controller/Adminhtml/Order/Create.php

$this->\_eventManager->dispatch('adminhtml\_sales\_order\_create\_process\_data', $eventData);

* Event name: sales\_order\_place\_before
* File: vendor/magento/module-sales/Model/Order.php

$this->\_eventManager->dispatch('sales\_order\_place\_before', ['order' => $this]);

* Event name: sales\_order\_place\_after
* File: vendor/magento/module-sales/Model/Order.php

$this->\_eventManager->dispatch('sales\_order\_place\_after', ['order' => $this]);

* Event name: order\_cancel\_after
* File: vendor/magento/module-sales/Model/Order.php

$this->\_eventManager->dispatch('order\_cancel\_after', ['order' => $this]);

* Event name: sales\_convert\_order\_to\_quote
* File: vendor/magento/module-sales/Model/AdminOrder/Create.php

$this->\_eventManager->dispatch('sales\_convert\_order\_to\_quote', ['order' => $order, 'quote' => $quote]);

* Event name: sales\_convert\_order\_item\_to\_quote\_item
* File: vendor/magento/module-sales/Model/AdminOrder/Create.php

$this->\_eventManager->dispatch(

'sales\_convert\_order\_item\_to\_quote\_item',

['order\_item' => $orderItem, 'quote\_item' => $item]

);

* Event name: checkout\_submit\_all\_after
* File: vendor/magento/module-sales/Model/AdminOrder/Create.php

$this->\_eventManager->dispatch('checkout\_submit\_all\_after', ['order' => $order, 'quote' => $quote]);

* Event name: \_sales\_email\_general\_async\_sending\_
* File: vendor/magento/module-sales/Model/Config/Backend/Email/AsyncSending.php

$this->\_eventManager->dispatch(

$this->\_eventPrefix . '\_sales\_email\_general\_async\_sending\_' . $state,

$this->\_getEventData()

);

* Event name: \_dev\_grid\_async\_indexing\_
* File: vendor/magento/module-sales/Model/Config/Backend/Grid/AsyncIndexing.php

$this->\_eventManager->dispatch(

$this->\_eventPrefix . '\_dev\_grid\_async\_indexing\_' . $state,

$this->\_getEventData()

);

* Event name: sales\_order\_invoice\_pay
* File: vendor/magento/module-sales/Model/Order/Invoice.php

$this->\_eventManager->dispatch('sales\_order\_invoice\_pay', [$this->\_eventObject => $this]);

* Event name: sales\_order\_invoice\_cancel
* File: vendor/magento/module-sales/Model/Order/Invoice.php

$this->\_eventManager->dispatch('sales\_order\_invoice\_cancel', [$this->\_eventObject => $this]);

* Event name: sales\_order\_invoice\_register
* File: vendor/magento/module-sales/Model/Order/Invoice.php

$this->\_eventManager->dispatch(

'sales\_order\_invoice\_register',

[$this->\_eventObject => $this, 'order' => $order]

);

* Event name: sales\_order\_item\_cancel
* File: vendor/magento/module-sales/Model/Order/Item.php

$this->\_eventManager->dispatch('sales\_order\_item\_cancel', ['item' => $this]);

* Event name: sales\_order\_payment\_place\_start
* File: vendor/magento/module-sales/Model/Order/Payment.php

$this->\_eventManager->dispatch('sales\_order\_payment\_place\_start', ['payment' => $this]);

* Event name: sales\_order\_payment\_place\_end
* File: vendor/magento/module-sales/Model/Order/Payment.php

$this->\_eventManager->dispatch('sales\_order\_payment\_place\_end', ['payment' => $this]);

* Event name: sales\_order\_payment\_cancel\_invoice
* File: vendor/magento/module-sales/Model/Order/Payment.php

$this->\_eventManager->dispatch(

'sales\_order\_payment\_cancel\_invoice',

['payment' => $this, 'invoice' => $invoice]

);

* Event name: sales\_order\_payment\_void
* File: vendor/magento/module-sales/Model/Order/Payment.php

$this->\_eventManager->dispatch('sales\_order\_payment\_void', ['payment' => $this, 'invoice' => $document]);

* Event name: sales\_order\_payment\_refund
* File: vendor/magento/module-sales/Model/Order/Payment.php

$this->\_eventManager->dispatch(

'sales\_order\_payment\_refund',

['payment' => $this, 'creditmemo' => $creditmemo]

);

* Event name: sales\_order\_payment\_cancel\_creditmemo
* File: vendor/magento/module-sales/Model/Order/Payment.php

$this->\_eventManager->dispatch(

'sales\_order\_payment\_cancel\_creditmemo',

['payment' => $this, 'creditmemo' => $creditmemo]

);

* Event name: sales\_order\_payment\_cancel
* File: vendor/magento/module-sales/Model/Order/Payment.php

$this->\_eventManager->dispatch('sales\_order\_payment\_cancel', ['payment' => $this]);

* Event name: sales\_order\_status\_unassign
* File: vendor/magento/module-sales/Model/Order/Status.php

$this->\_eventManager->dispatch(

'sales\_order\_status\_unassign',

[

'status' => $this->getStatus(),

'state' => $state

]

);

* Event name: \_html\_txn\_id
* File: vendor/magento/module-sales/Model/Order/Payment/Transaction.php

$this->\_eventManager->dispatch($this->\_eventPrefix . '\_html\_txn\_id', $this->\_getEventData());

* Event name: \_load\_after
* File: vendor/magento/module-sales/Model/ResourceModel/Order/Address/Collection.php

$this->\_eventManager->dispatch($this->\_eventPrefix . '\_load\_after', [$this->\_eventObject => $this]);

* Event name: \_set\_sales\_order
* File: vendor/magento/module-sales/Model/ResourceModel/Order/Collection/AbstractCollection.php

$this->\_eventManager->dispatch(

$this->\_eventPrefix . '\_set\_sales\_order',

['collection' => $this, $this->\_eventObject => $this, 'order' => $order]

);

* Event name: sales\_sale\_collection\_query\_before
* File: vendor/magento/module-sales/Model/ResourceModel/Sale/Collection.php

$this->\_eventManager->dispatch('sales\_sale\_collection\_query\_before', ['collection' => $this]);

* Event name: on\_view\_report
* File: vendor/magento/module-search/Controller/Adminhtml/Term/Report.php

$this->\_eventManager->dispatch('on\_view\_report', ['report' => 'search']);

* Event name: sendfriend\_product
* File: vendor/magento/module-send-friend/Controller/Product/Send.php

$this->\_eventManager->dispatch('sendfriend\_product', ['product' => $product]);

* Event name: swatch\_gallery\_upload\_image\_after
* File: vendor/magento/module-swatches/Controller/Adminhtml/Iframe/Show.php

$this->\_eventManager->dispatch(

'swatch\_gallery\_upload\_image\_after',

['result' => $result, 'action' => $this]

);

* Event name: adminhtml\_cache\_refresh\_type
* File: vendor/magento/module-tax/Controller/Adminhtml/Tax/IgnoreTaxNotification.php

$this->\_eventManager->dispatch('adminhtml\_cache\_refresh\_type', ['type' => 'block\_html']);

* Event name: tax\_rate\_data\_fetch
* File: vendor/magento/module-tax/Model/Calculation.php

$this->\_eventManager->dispatch('tax\_rate\_data\_fetch', ['request' => $request, 'sender' => $this]);

* Event name: tax\_settings\_change\_after
* File: vendor/magento/module-tax/Model/Calculation/Rate.php

$this->\_eventManager->dispatch('tax\_settings\_change\_after');

* Event name: page\_block\_html\_topmenu\_gethtml\_before
* File: vendor/magento/module-theme/Block/Html/Topmenu.php

$this->\_eventManager->dispatch(

'page\_block\_html\_topmenu\_gethtml\_before',

['menu' => $this->\_menu, 'block' => $this]

);

* Event name: page\_block\_html\_topmenu\_gethtml\_after
* File: vendor/magento/module-theme/Block/Html/Topmenu.php

$this->\_eventManager->dispatch(

'page\_block\_html\_topmenu\_gethtml\_after',

['menu' => $this->\_menu, 'transportObject' => $transportObject]

);

* Event name: assign\_theme\_to\_stores\_after
* File: vendor/magento/module-theme/Model/Config.php

$this->\_eventManager->dispatch(

'assign\_theme\_to\_stores\_after',

['stores' => $stores, 'scope' => $scope, 'theme' => $theme]

);

* Event name: permissions\_role\_html\_before
* File: vendor/magento/module-user/Block/Role.php

$this->\_eventManager->dispatch('permissions\_role\_html\_before', ['block' => $this]);

* Event name: admin\_permissions\_role\_prepare\_save
* File: vendor/magento/module-user/Controller/Adminhtml/User/Role/SaveRole.php

$this->\_eventManager->dispatch(

'admin\_permissions\_role\_prepare\_save',

['object' => $role, 'request' => $this->getRequest()]

);

* Event name: admin\_user\_authenticate\_before
* File: vendor/magento/module-user/Model/User.php

$this->\_eventManager->dispatch(

'admin\_user\_authenticate\_before',

['username' => $username, 'user' => $this]

);

* Event name: admin\_user\_authenticate\_after
* File: vendor/magento/module-user/Model/User.php

$this->\_eventManager->dispatch(

'admin\_user\_authenticate\_after',

['username' => $username, 'password' => $password, 'user' => $this, 'result' => $result]

);

* Event name: admin\_user\_authenticate\_after
* File: vendor/magento/module-user/Model/User.php

$this->\_eventManager->dispatch(

'admin\_user\_authenticate\_after',

[

'username' => $this->getUserName(),

'password' => $passwordString,

'user' => $this,

'result' => $isCheckSuccessful

]

);

* Event name: product\_option\_renderer\_init
* File: vendor/magento/module-wishlist/Block/Customer/Wishlist/Item/Options.php

$this->\_eventManager->dispatch('product\_option\_renderer\_init', ['block' => $this]);

* Event name: wishlist\_add\_product
* File: vendor/magento/module-wishlist/Controller/Index/Add.php

$this->\_eventManager->dispatch(

'wishlist\_add\_product',

['wishlist' => $wishlist, 'product' => $product, 'item' => $result]

);

* Event name: wishlist\_share
* File: vendor/magento/module-wishlist/Controller/Index/Send.php

$this->\_eventManager->dispatch('wishlist\_share', ['wishlist' => $wishlist]);

* Event name: wishlist\_items\_renewed
* File: vendor/magento/module-wishlist/Helper/Data.php

$this->\_eventManager->dispatch('wishlist\_items\_renewed');

* Event name: wishlist\_product\_add\_after
* File: vendor/magento/module-wishlist/Model/Wishlist.php

$this->\_eventManager->dispatch('wishlist\_product\_add\_after', ['items' => $items]);

* Event name: wishlist\_item\_collection\_products\_after\_load
* File: vendor/magento/module-wishlist/Model/ResourceModel/Item/Collection.php

$this->\_eventManager->dispatch(

'wishlist\_item\_collection\_products\_after\_load',

['product\_collection' => $productCollection]

);

## JavaScript Varien Events

* Event name: formSubmit
* File: lib/web/mage/adminhtml/form.js

varienGlobalEvents.fireEvent("formSubmit", $(e.target).attr('id'));

* Event name: address\_country\_changed
* File: lib/web/mage/adminhtml/form.js

varienGlobalEvents.fireEvent("address\_country\_changed", this.countryEl);

* Event name: gridRowClick
* File: lib/web/mage/adminhtml/grid.js

varienGlobalEvents.fireEvent('gridRowClick', event);

* Event name: gridRowDblClick
* File: lib/web/mage/adminhtml/grid.js

varienGlobalEvents.fireEvent('gridRowDblClick', event);

* Event name: tinymceChange
* File: lib/web/mage/adminhtml/wysiwyg/widget.js

varienGlobalEvents.fireEvent('tinymceChange');

* Event name: tinymceSubmit
* File: lib/web/mage/adminhtml/wysiwyg/tiny\_mce/setup.js

varienGlobalEvents.fireEvent('tinymceSubmit', e);

* Event name: tinymcePaste
* File: lib/web/mage/adminhtml/wysiwyg/tiny\_mce/setup.js

varienGlobalEvents.fireEvent('tinymcePaste', o);

* Event name: tinymceBeforeSetContent
* File: lib/web/mage/adminhtml/wysiwyg/tiny\_mce/setup.js

varienGlobalEvents.fireEvent('tinymceBeforeSetContent', o);

* Event name: tinymceSetContent
* File: lib/web/mage/adminhtml/wysiwyg/tiny\_mce/setup.js

varienGlobalEvents.fireEvent('tinymceSetContent', o);

* Event name: tinymceSetContent
* File: lib/web/mage/adminhtml/wysiwyg/tiny\_mce/setup.js

varienGlobalEvents.fireEvent('tinymceSaveContent', o);

* Event name: tinymceChange
* File: lib/web/mage/adminhtml/wysiwyg/tiny\_mce/setup.js

varienGlobalEvents.fireEvent('tinymceChange', l);

* Event name: tinymceExecCommand
* File: lib/web/mage/adminhtml/wysiwyg/tiny\_mce/setup.js

varienGlobalEvents.fireEvent('tinymceExecCommand', cmd);

* Event name: open\_browser\_callback
* File: lib/web/mage/adminhtml/wysiwyg/tiny\_mce/setup.js

varienGlobalEvents.fireEvent("open\_browser\_callback", {

win: w,

type: objectType,

field: fieldName

});