-Magento 2 Plugin là 1 plugin kĩ thuật giúp viết code tốt hơn. Interceptor Plugin là 1 tiện ích mở rộng nhỏ của Magento 2 cho phép chỉnh sửa hành vi của bất kì lớp hoặc phương thức công khai nào bằng cách chặn một lệnh gọi hàm và chạy mã before, after, around lệnh gọi hàm. Bằng cách sử dụng Magento 2 Plugin Interceptor, bạn có thể sửa đổi hành vi của một lớp trong khi không cần phải thay đổi trực tiếp lớp đó

-Thoạt nhìn qua, có vẻ như observers có thể giúp chúng ta làm được điều tương tự. Tuy nhiên, có một số điểm khác biệt cần phải được chỉ rõ ở đây. Interceptor plugin có thể xác nhận với 1 sortOrder cho phép kiểm tra thứ tự mà các plugin chạy. Đó là lý do plugin không làm thay đổi class và không có bất kì xung đột nào với nhau

-Developer có thể:

+Chuyển tiếp bất kỳ lệnh gọi phương thức nào được sử dụng trên đối tượng do trình quản lý đối tượng kiểm soát và thực hiện hành động theo chương trình

+Sửa đổi giá trị trả về của bất kỳ lệnh gọi phương thức nào được sử dụng trên đối tượng do trình quản lý đối tượng kiểm soát

+Sửa đổi đối số của bất kỳ lệnh gọi phương thức nào được sử dụng trên đối tượng do trình quản lý đối tượng kiểm soát

+Tiến hành tương tự khi các mô-đun khác đang tiến hành cùng một phương thức theo cùng một cách hoặc có thể dự đoán được.

-Các options mà Plugin không thể làm việc cùng:

+Objects được khởi tạo trước khi Magento\Framework\Interception được khởi động

+Final methods

+Final classess

+class có chứa ít nhất 1 final public method

+non-public methods

+\_\_construct

+Virtual types

-Các bước tạo 1 plugin mới:

+Khai báo 1 plugin: tạo file app/code/{namespace}/{module}/etc/di.xml với nội dung như sau:

<config>

<type name="{ObservedType}">

<plugin name="{pluginName}" type="{PluginClassName}" sortOrder="1" disabled="false"/>

</type>

</config>

\*Trong đó:

+type name: tên của 1 class hoặc 1 interface cần follow

+plugin name: tên tùy ý cho plugin

+plugin type: điền tên của plugin’s class hoặc virtual type của nó, VD: \Vendor\Module\Plugin\<ModelName>Plugin

+plugin sortOrder (optional): cài đặt thứ tự khi plugin gọi với các mtethods tương tự khác trong quá trình

+plugin disabled (optional): cho phép bật tắt plugin nhanh chóng, mặc định false. Sử dụng để tắt core hoặc third-party plugins trong di.xml

VD: tạo 1 file app\code\Mageplaza\HelloWorld\etc\di.xml như sau:

<config xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="../../../../../lib/internal/Magento/Framework/ObjectManager/etc/config.xsd">

<type name="Mageplaza\HelloWorld\Controller\Index\Example">

<plugin name="Mageplaza\_HelloWorld\_Plugin" type="Mageplaza\HelloWorld\Plugin\ExamplePlugin" sortOrder="10" disabled="false" />

</type>

</config>

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

<?php

namespace Mageplaza\HelloWorld\Controller\Index;

class Example extends \Magento\Framework\App\Action\Action

{

protected $title;

public function execute()

{

echo $this->setTitle('Welcome');

echo $this->getTitle();

}

public function setTitle($title)

{

return $this->title = $title;

}

public function getTitle()

{

return $this->title;

}

}

Với plugin name, tạo 1 file Example.php trong app/code/Mageplaza/HelloWorld/Plugin/ với nội dung:

<?php

namespace Mageplaza\HelloWorld\Plugin;

class ExamplePlugin{

}

-1 Plugin là 1 cách tuyệt vời để mở rộng hoặc chỉnh sửa hành vi của public method bằng cách sử dụng before/after/around. Đầu tiên, có 1 object có quyền với mọi public methods của observerd method’s class

-3 methods in plugin:

+before – beforeDispatch() – chạy trước observed method

+around – arounDispatch() – chạy ngay sau observed method

+after – afterDispatch() – chạy cả trước và sau observed method

VD:

<?php

namespace Mageplaza\HelloWorld\Plugin;

class ExamplePlugin

{

public function beforeSetTitle(\Mageplaza\HelloWorld\Controller\Index\Example $subject, $title)

{

$title = $title . " to ";

echo \_\_METHOD\_\_ . "</br>";

return [$title];

}

public function afterGetTitle(\Mageplaza\HelloWorld\Controller\Index\Example $subject, $result)

{

echo \_\_METHOD\_\_ . "</br>";

return '<h1>'. $result . 'Mageplaza.com' .'</h1>';

}

public function aroundGetTitle(\Mageplaza\HelloWorld\Controller\Index\Example $subject, callable $proceed)

{

echo \_\_METHOD\_\_ . " - Before proceed() </br>";

$result = $proceed();

echo \_\_METHOD\_\_ . " - After proceed() </br>";

return $result;

}

}