Adobe.Premium.AD0-E716.69q - DEMO

Number: AD0-E716
Passing Score: 800
Time Limit: 120 min
File Version: 1.7

Exam Code: AD0-E716

Website: www.VCEplus.io
Twitter: www.twitter.com/VCE_Plus

Exam A

QUESTION 1

An Adobe Commerce developer has added an iframe and included a JavaScript library from an external domain to the website. After that, they found the following error in the console:

Refused to frame [URL] because it violates the Content Security Policy directive.

In order to fix this error, what would be the correct policy ids to add to the csp_whitelist.xml file?

A. frame-src and script-src

B. default-src and object-src

C. frame-ancestors and connect-src

Correct Answer: C
Section:
Explanation:

The frame-ancestors directive specifies the domains that are allowed to embed the current page in an iframe. The connect-src directive specifies the domains that are allowed to be loaded by the current page through a <script> tag or XMLHttpRequest.

In this case, the developer has added an iframe that embeds a page from an external domain. The Content Security Policy (CSP) is preventing the iframe from being loaded because the domain of the external page is not listed in the frame-ancestors directive.

To fix this error, the developer needs to add the domain of the external page to the frame-ancestors directive. They can do this by adding the following line to the csp_whitelist.xml file:

<frame-ancestors>https://www.example.com</frame-ancestors>

QUESTION 2

An Adobe Commerce Developer is tasked with creating a custom form which submits its data to a frontend controller They have decided to create an action and have implemented the \Magento\Framework\App \Action\HttpPostActioninterface class, but are not seeing the data being persisted in the database, and an error message is being shown on the frontend after submission.

After debugging and ensuring that the data persistence logic is correct, what may be cause and solution to this?

A. Magento does not allow POST requests to a frontend controller, therefore, the submission functionality will need to be rewritten as an API endpoint.

B. The developer forgot to implement a validatePostDataQ method in their action. They should implement this method: all non-validated POST data gets stripped out of the request and an error is thrown.

C. Form key validation runs on all non-AJAX POST requests, the developer needs to add the for_key to their requests.

Correct Answer: C
Section:
Explanation:

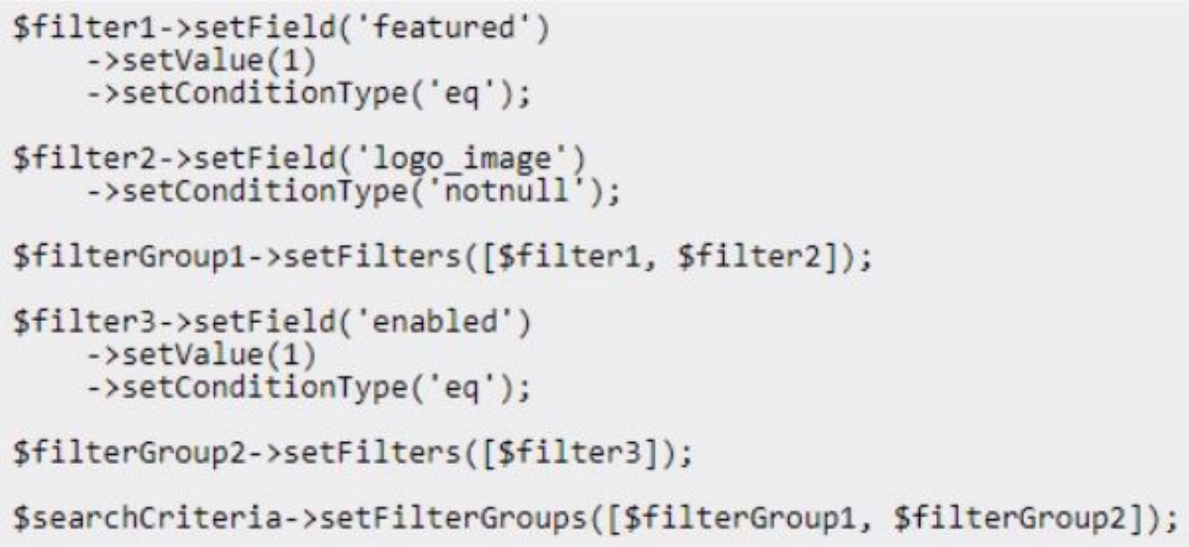
According to the Magento Stack Exchange answer, form key validation is a security feature that prevents CSRF attacks by checking if the form key in the request matches the one generated by Magento. If the developer does not include the form_key in their custom form, the validation will fail and an error will be shown. Therefore, the developer needs to add the form_key to their requests by using <?= $block-> getBlockHtml ('formkey') ?> in their template file. Verified

Reference: https://magento.stackexchange.com/questions/95171/magento-2-form-validation

QUESTION 3

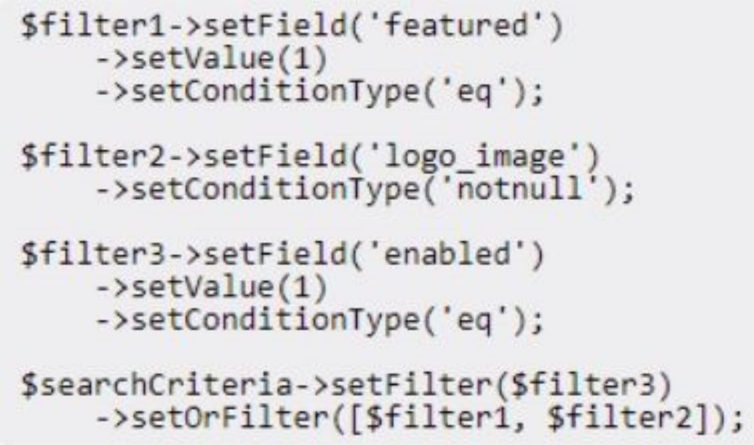
An Adobe Commerce developer is working on a module to manage custom brand entities and wants to replicate the following SQL query using SearchCriteria:

A)



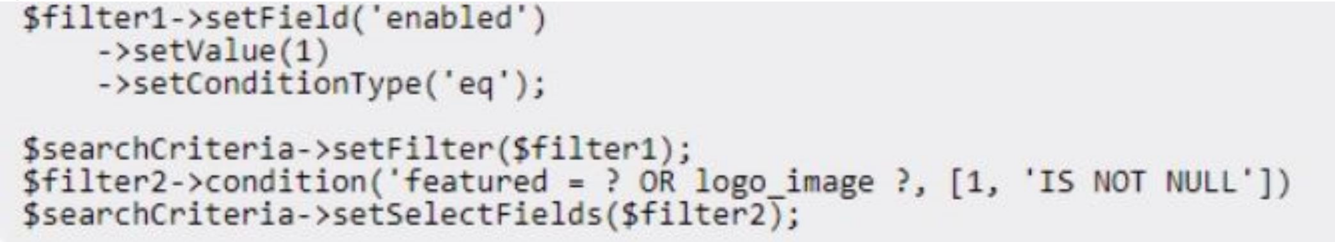
```
$filter1->setField('featured')
    ->setValue(1)
    ->setConditionType('eq');

$filter2->setField('logo_image')
    ->setConditionType('notnull');

$filterGroup1->setFilters([$filter1, $filter2]);

$filter3->setField('enabled')
    ->setValue(1)
    ->setConditionType('eq');

$filterGroup2->setFilters([$filter3]);

$searchCriteria->setFilterGroups([$filterGroup1, $filterGroup2]);
```

B)



```
$filter1->setField('featured')
    ->setValue(1)
    ->setConditionType('eq');

$filter2->setField('logo_image')
    ->setConditionType('notnull');

$filter3->setField('enabled')
    ->setValue(1)
    ->setConditionType('eq');

$searchCriteria->setFilter($filter3)
    ->setOrFilter([$filter1, $filter2]);
```

C)

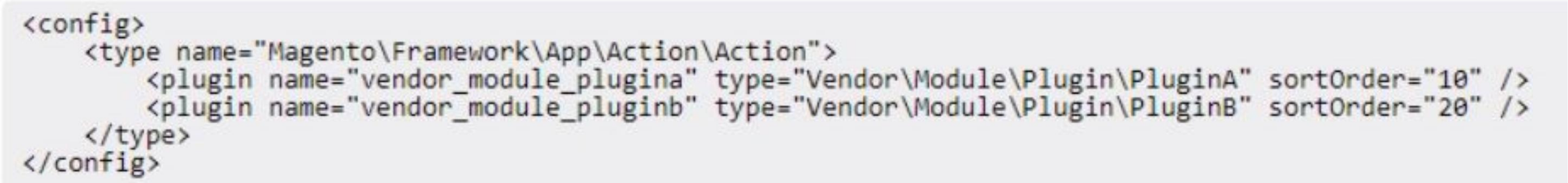


```
$filter1->setField('enabled')
    ->setValue(1)
    ->setConditionType('eq');

$searchCriteria->setFilter($filter1);
$filter2->condition('featured = ? OR logo_image ?, [1, 'IS NOT NULL'])
$searchCriteria->setSelectFields($filter2);
```

A. Option A
B. Option B
C. Option C

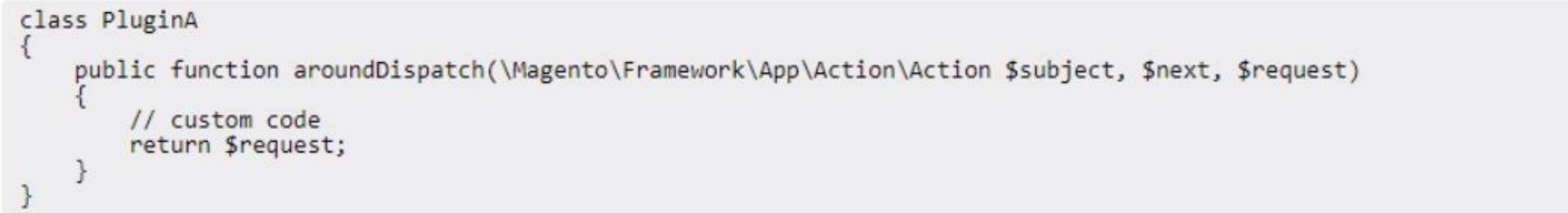
Correct Answer: B
Section:
Explanation:
The following SearchCriteria query will replicate the SQL query:
$searchCriteria = new \Magento\Framework\Api\SearchCriteriaBuilder();
$searchCriteria->addFilter('name', 'Brand 1', 'eq');
$searchCriteria->addFilter('status', 1, 'eq');
$brandCollection = $this->brandRepository->getList($searchCriteria);

QUESTION 4
The di. xml file of a module attaches two plugins for the class Action.



```
<config>
    <type name="Magento\Framework\App\Action\Action">
        <plugin name="vendor_module_plugina" type="Vendor\Module\Plugin\PluginA" sortOrder="10" />
        <plugin name="vendor_module_pluginb" type="Vendor\Module\Plugin\PluginB" sortOrder="20" />
    </type>
</config>
```

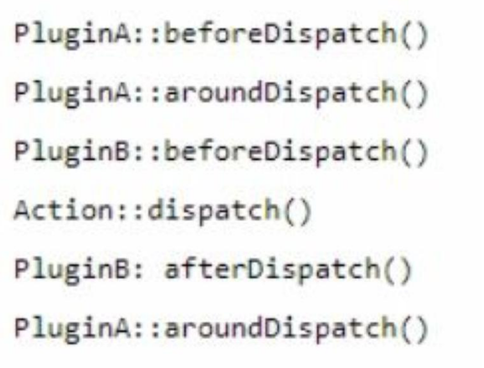
The around plugin code is:



```
class PluginA
{
    public function aroundDispatch(\Magento\Framework\App\Action\Action $subject, $next, $request)
    {
        // custom code
        return $request;
    }
}
```

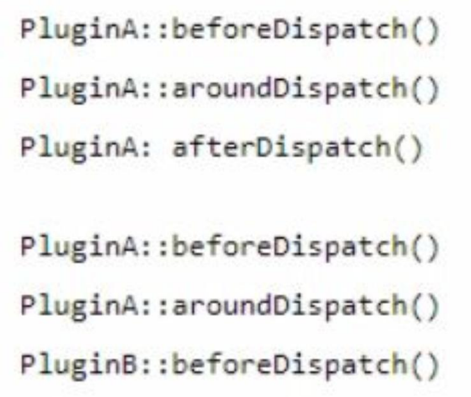
What would be the plugin execution order?

A)



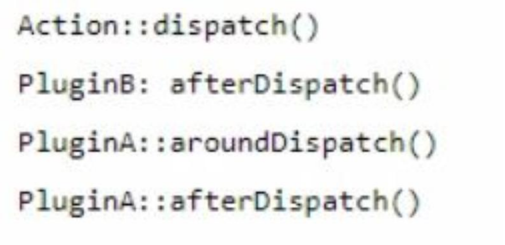
```
PluginA::beforeDispatch()
PluginA::aroundDispatch()
PluginB::beforeDispatch()
Action::dispatch()
PluginB: afterDispatch()
PluginA::aroundDispatch()
```

B)



```
PluginA::beforeDispatch()
PluginA::aroundDispatch()
PluginA: afterDispatch()

PluginA::beforeDispatch()
PluginA::aroundDispatch()
PluginB::beforeDispatch()
```

C)



```
Action::dispatch()
PluginB: afterDispatch()
PluginA::aroundDispatch()
PluginA::afterDispatch()
```

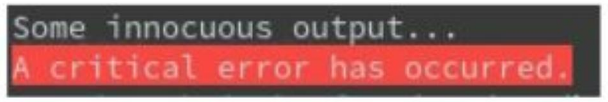
A. Option A
B. Option B
C. Option C

Correct Answer: C
Section:
Explanation:
The plugin execution order is as follows:
PluginA::beforeDispatch()
PluginB::beforeDispatch()
PluginA::aroundDispatch()
The code in the around plugin
PluginB::afterDispatch()
PluginA::afterDispatch()
The aroundDispatch() method is executed in a separate scope, so the code in the around plugin will be executed after the beforeDispatch() methods of both plugins, but before the afterDispatch() methods of both plugins.
Here is a diagram that shows the plugin execution order:
PluginA
beforeDispatch()
aroundDispatch()
afterDispatch()
PluginB
beforeDispatch()
afterDispatch()

QUESTION 5
An Adobe Commerce developer adds a new extension attribute to add an array of values to the invoices that are fetched through the APIs.
After a while, their technical manager reviews their work and notices something wrong with the extension_attributes. xml file that the developer created in their module:
What is the problem with this xml snippet?
A. The extension attribute references the wrong interface, it should have referenced the Magento\saies\Api\data\invoiceinterface.
B. The extension attribute references the repository instead of the interface it implements (Magento\saies\Api\invoiceRepositorymterface).
C. The type is wrong, string [] should be replaced with array.

Correct Answer: B
Section:
Explanation:
The extension attribute is referencing the repository instead of the interface it implements. The correct XML snippet should be:
XML
<extension_attributes>
group='General'
translate='true'>
<description>This attribute stores an array of custom values for the invoice.</description>
<source_model>Magento\Sales\Api\Data\InvoiceInterface</source_model>
</extension_attributes>
The source_model attribute specifies the interface that the extension attribute is associated with. In this case, the extension attribute is associated with the Magento\Sales\Api\Data\InvoiceInterface interface.

QUESTION 6
An Adobe Commerce developer is creating a new console command to perform a complex task with a lot of potential terminal output. If an error occurs, they want to provide a message that has higher visibility than some of the other content that may be appearing, so they want to ensure it is highlighted in red (as seen in the screenshot):



```
Some innocuous output...
A critical error has occurred.
```

How can they customize the appearance of this message?

A. Call the setDecorationType(Stype) method On the Symfony\Console\Output\OutputInterface Object before Calling writeln().

B. Wrap the output content in tags like <error>, <info>, or <comment>.

C. Throw a new commandException with the desired message passed as an argument.

Correct Answer: A
Section:
Explanation:

The developer can customize the appearance of the error message by calling the setDecorationType() method on the Symfony\Console\Output\OutputInterface object before calling writeln(). The setDecorationType() method takes a single argument, which is the type of decoration that the developer wants to use. In this case, the developer wants to use the STYPE_ERROR decoration, which will highlight the message in red.

Here is an example of how to customize the appearance of the error message:

```
$output = new Symfony\Console\Output\ConsoleOutput();
$output->setDecorationType(Symfony\Console\Output\OutputInterface::STYPE_ERROR);
$output->writeln('This is an error message.');
```

The output of this code will be an error message that is highlighted in red.

QUESTION 7

An Adobe Commerce developer is being tasked with creating a new cron job to run a method that has already been written. What are the minimally required steps to accomplish this?

A. Create a crontab.xmi file and a new system configuration in system.xmi for the schedule.

B. Create crontab.xmi and cron_groups.xmi files to assign the new job to a cron group.

C. Create a crontab.xmi file and set a schedule for the new cron job.

Correct Answer: C
Section:
Explanation:

According to the Configure and run cron guide for Magento 2 developers, the crontab.xmi file is used to declare and configure cron jobs for a module. The file should specify the name, instance, method and schedule of the cron job. Therefore, creating a crontab.xmi file and setting a schedule for the new cron job are the minimally required steps to accomplish this task. Verified

Reference: https://devdocs.magento.com/guides/v2.3/config-guide/cli/config-cli-subcommands-cron.html

QUESTION 8

Which hashing algorithm will Adobe Commerce choose to hash customer passwords?

A. If the Sodium extension is installed, SHA256 will be chosen, otherwise MD5 will be used as the Magento default hashing algorithm.

B. If the Sodium extension is installed, Argon 2ID13 will be chosen, otherwise SHA256 will be used as the Magento default hashing algorithm.

C. It does not matter if the Sodium extension is installed or not, the Magento hashing default algorithm will be SHA256.

Correct Answer: B
Section:
Explanation:

If the Sodium extension is installed, Argon 2ID13 will be chosen as the Magento default hashing algorithm. Otherwise, SHA256 will be used.

The Sodium extension is a PHP extension that provides cryptographic functions. Argon 2ID13 is a password hashing algorithm that is considered to be more secure than SHA256.

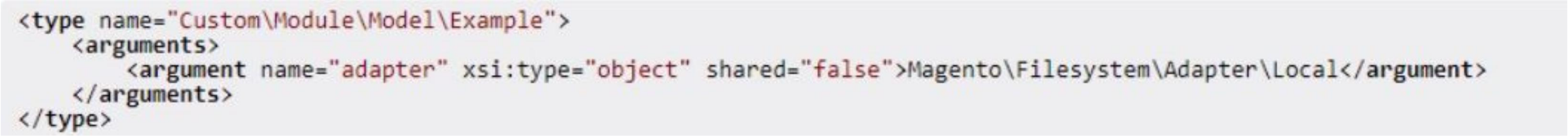
If the Sodium extension is installed, Magento will use Argon 2ID13 as the default hashing algorithm for customer passwords. If the Sodium extension is not installed, Magento will use SHA256 as the default hashing algorithm.

QUESTION 9

An Adobe Commerce developer is developing a custom module. As part of their implementation they have decided that all instances of their Custom\Module\Model\Example class should receive a new instance of Magento\Filesystem\Adapter\Local.

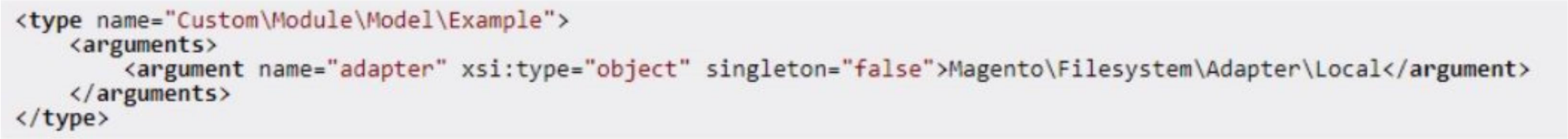
How would the developer achieve this using di. xml?

A)



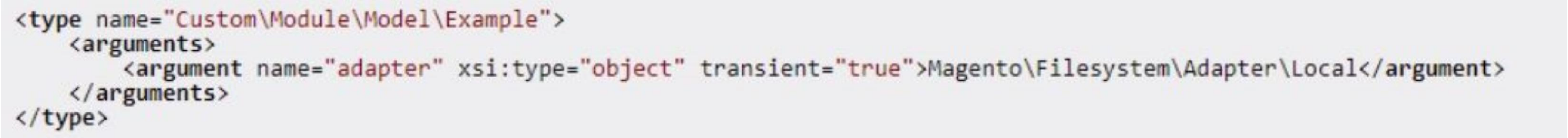
```
<type name="Custom\Module\Model\Example">
    <arguments>
        <argument name="adapter" xsi:type="object" shared="false">Magento\Filesystem\Adapter\Local</argument>
    </arguments>
</type>
```

B)



```
<type name="Custom\Module\Model\Example">
    <arguments>
        <argument name="adapter" xsi:type="object" singleton="false">Magento\Filesystem\Adapter\Local</argument>
    </arguments>
</type>
```

C)



```
<type name="Custom\Module\Model\Example">
    <arguments>
        <argument name="adapter" xsi:type="object" transient="true">Magento\Filesystem\Adapter\Local</argument>
    </arguments>
</type>
```

A. Option A
B. Option B
C. Option C

Correct Answer: B
Section:
Explanation:
The developer can achieve this by adding the following configuration to their di.xml file:
XML
<config>
<component name='Custom\Module\Model\Example' factory='Custom\Module\Model\ExampleFactory'>
</component>
</config>
This configuration will ensure that all instances of the Custom\Module\Model\Example class will receive a new instance of the Magento\Filesystem\Adapter\Local class.

QUESTION 10

An Adobe Commerce developer has been tasked with applying a pricing adjustment to products on the website. The adjustments come from a database table. In this case, catalog price rules do not work. They created a plugin for getPrice on the price model, but the layered navigation is still displaying the old price.
How can this be resolved?
A. Create an implementation for \Magento\Catalog\Hodel\Product\PriceModifierInterf ace.
B. Create an after plugin On \Magento\Catalog\Api\Data\BasePriceInterface:: getPrice.
C. Create a plugin for\Magento\Catalog\Model\Indexer\Product\Price::executeRow.

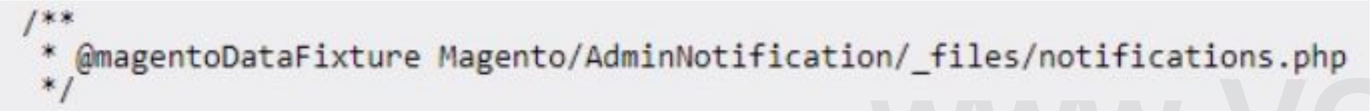
Correct Answer: C
Section:
Explanation:
The developer can resolve this issue by creating a plugin for the Magento\Catalog\Model\Indexer\Product\Price::executeRow() method. This method is responsible for updating the product price index.
The plugin can be used to add the pricing adjustment from the database to the product price index. Once the product price index is updated, the layered navigation will display the correct price.
Here is an example of a plugin for the executeRow() method:
PHP
class MyPlugin
{
public function executeRow(

```
\Magento\Catalog\Model\Indexer\Product\Price $subject,
\Magento\Catalog\Model\Product $product,
array $data
) {
$adjustment = $this->getAdjustment($product);
$product->setPrice($product->getPrice() + $adjustment);
}
private function getAdjustment(Product $product)
{
$adjustment = $product->getData('adjustment');
if (!is_numeric($adjustment)) {
return 0;
}
return $adjustment;
}
}
```

This plugin will add the adjustment data from the product to the product price index. Once the product price index is updated, the layered navigation will display the correct price.

QUESTION 11

An Adobe Commerce developer is writing an integration test. They checked some Integration Tests for Magento core modules for reference and noticed that they use data fixtures initialized by adding annotations to test classes. For example:



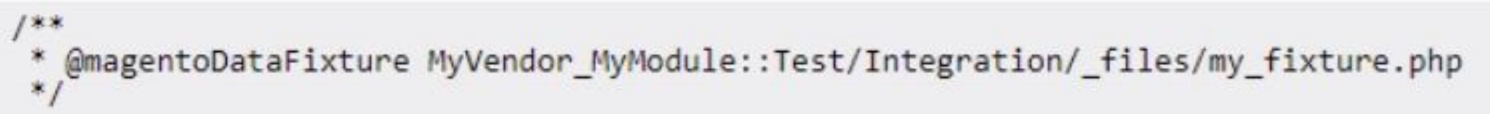
```
/**
 * @magentoDataFixture Magento/AdminNotification/_files/notifications.php
 */
```

The developer wants to add their own fixture to test a MyVendor_MyModule they created. Which steps will make this possible?

A.

1- Create a PHP file with the fixture data inside their own module in [module dir]/Test/integration/_fiies/my_fixture.php.

2- Add the following annotation to the test method:

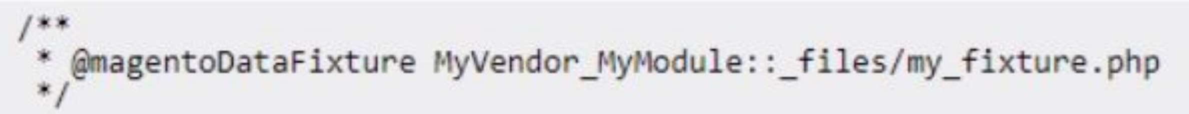


```
/**
 * @magentoDataFixture MyVendor_MyModule::Test/Integration/_files/my_fixture.php
 */
```

B.

1- Create a PHP file With the fixture data in [magento root dir]/dev/tests/integration/testsuite/MyVendor/MyModule/_files/my_fixture.php.

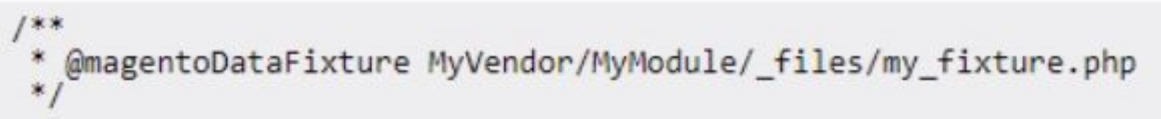
2- Add the following annotation to the test method:



```
/**
 * @magentoDataFixture MyVendor_MyModule::_files/my_fixture.php
 */
```

C. 1- Create a PHP file with the fixture data inside their own module in [module dir]/Test/integration/_f iies/my_f ixture.php.

2- Add the following annotation to the test method:



```
/**
 * @magentoDataFixture MyVendor/MyModule/_files/my_fixture.php
 */
```

Correct Answer: B
Section:
Explanation:

To add a custom fixture to test a MyVendor_MyModule, the developer needs to do the following:

Create a PHP file with the fixture data in[magento root dir]/dev/tests/integration/testsuite/MyVendor/MyModule/_files/my_fixture.php.
Add the following annotation to the test method:
@magentoDataFixture(
'testsuite/MyVendor/MyModule/_files/my_fixture.php'
)
This will tell Magento to load the fixture data from the my_fixture.php file before the test method is executed.

QUESTION 12
An Adobe Commerce developer has created a before plugin for the save() function within the Magento\Framework\App\cache\Proxy class. The purpose of this plugin is to add a prefix on all cache identifiers that fulfill certain criteria.
Why is the plugin not executing as expected?
A. Another around plugin defined for the same function does not call the callable.
B. Cache identifiers are immutable and cannot be changed.
C. The target ClaSS implements Magento\Framework\ObjectManager\NoninterceptableInterface.

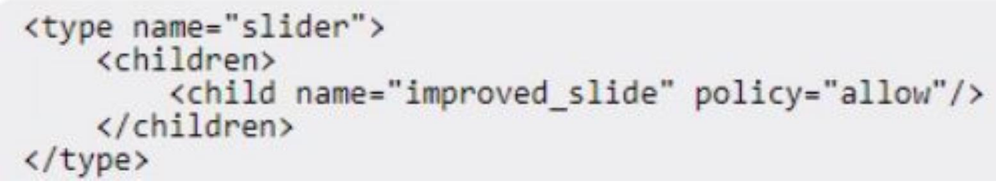
Correct Answer: C
Section:
Explanation:
According to the Plugins (Interceptors) guide for Magento 2 developers, plugins are class methods that modify the behavior of public class methods by intercepting them and running code before, after, or around them. However, some classes in Magento 2 implement the NoninterceptableInterface interface, which prevents plugins from being generated for them. The Magento\Framework\App\cache\Proxy class is one of them, as it extends from Magento\Framework\ObjectManager\NoninterceptableInterface. Therefore, the plugin is not executing as expected because the target class implements NoninterceptableInterface. Verified Reference: https://devdocs.magento.com/guides/v2.3/extension-dev-guide/plugins.html

QUESTION 13
An Adobe Commerce developer has installed a module from a third-party vendor. This module fires a custom event named third_party_event_after and also defines an observer named third_party_event_after_observer that listens to that event. The developer wants to listen to this custom event in their own module but wants to execute their observer's logic after the third_party_event_after_observer observer has finished executing.

What would the developer do to ensure their observer runs after the observer defined by the third-party module?
A. Ensure the third-party module is listed in the <sequence> node of the developer's module.xmi file.
B. Set the sort order of the new observer to be less than that of the third-party vendor's observer.
C. This is not possible as observers listening to the same event may be invoked in any order.

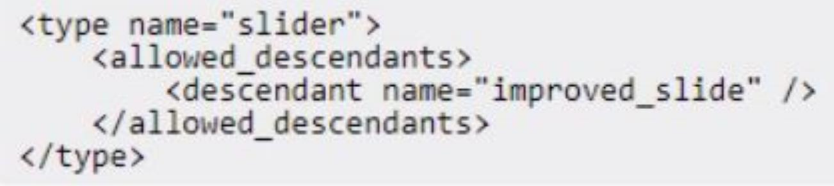
Correct Answer: B
Section:
Explanation:
To ensure that the developer's observer runs after the observer defined by the third-party module, they need to set the sort order of the new observer to be less than that of the third-party vendor's observer.
The sort order is a number that is assigned to each observer. The lower the number, the earlier the observer will be executed.
For example, if the third-party vendor's observer has a sort order of 10, the developer's observer would need to have a sort order of 9 or lower.

QUESTION 14
An Adobe Commerce developer has been asked to modify the PageBuilder slider content type to allow a new custom content type (other than slide) to be assigned as a child. The developer has already created the new content type called improved_slide in their module. They now need to create a new view/adminhtml/pagebuilder/content_type/slider. xml file in their module to allow the new content type to be a child of slider content types.
What is the correct xml to accomplish this?
A)



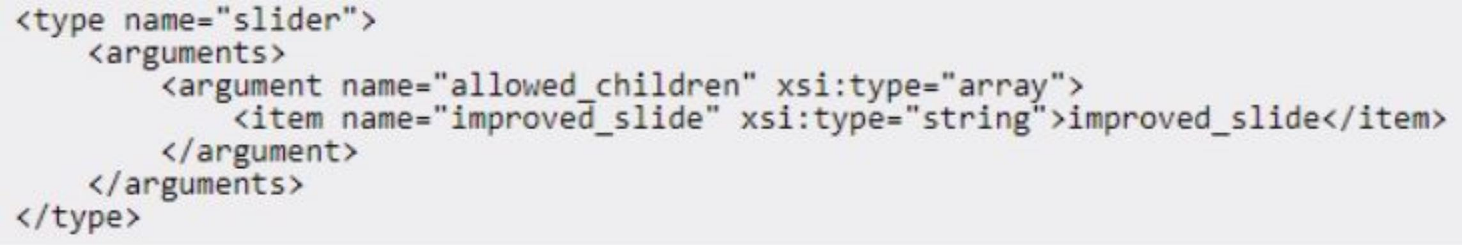
```
<type name="slider">
    <children>
        <child name="improved_slide" policy="allow"/>
    </children>
</type>
```

B)



```
<type name="slider">
    <allowed_descendants>
        <descendant name="improved_slide" />
    </allowed_descendants>
</type>
```

C)



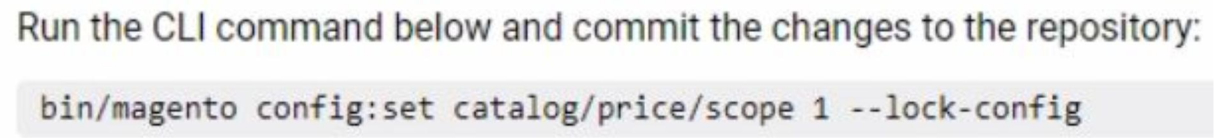
```
<type name="slider">
    <arguments>
        <argument name="allowed_children" xsi:type="array">
            <item name="improved_slide" xsi:type="string">improved_slide</item>
        </argument>
    </arguments>
</type>
```

A. Option A
B. Option B
C. Option C

Correct Answer: B
Section:
Explanation:
The following XML will allow the new content type to be a child of slider content types:
<type>slider</type>
<children>
<type>improved_slide</type>
</children>
Use code with caution.https://bard.google.com/faq
This XML will tell Magento that the slider content type can have improved_slide content types as children.

QUESTION 15
An Adobe Commerce developer creates a new website using a data patch. Each website will have unique pricing by website. The developer does not have visibility into the production and staging environments so they do not know what the configuration currently is.
How would they ensure the configuration is deployed and consistent across all environments?

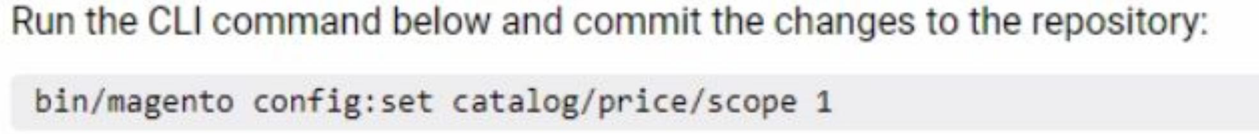
A.



Run the CLI command below and commit the changes to the repository:

```
bin/magento config:set catalog/price/scope 1 --lock-config
```

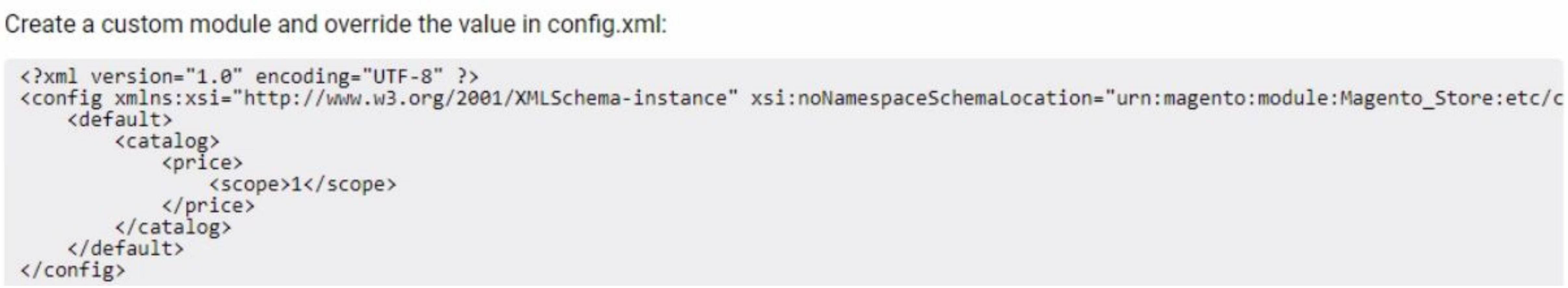
B.



Run the CLI command below and commit the changes to the repository:

```
bin/magento config:set catalog/price/scope 1
```

C.



Create a custom module and override the value in config.xml:

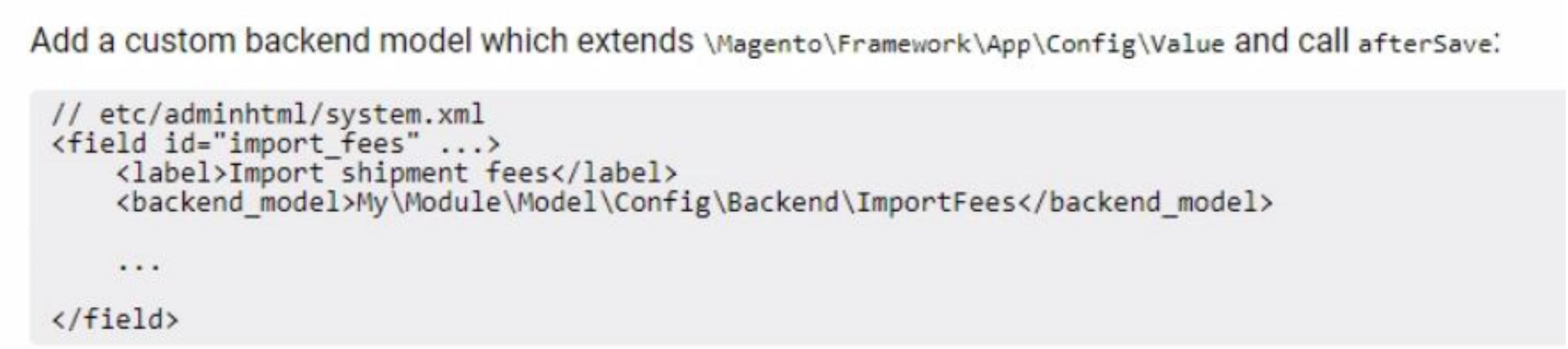
```
<?xml version="1.0" encoding="UTF-8" ?>
<config xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="urn:magento:module:Magento_Store:etc/c
    <default>
        <catalog>
            <price>
                <scope>1</scope>
            </price>
        </catalog>
    </default>
</config>
```

Correct Answer: B
Section:
Explanation:
To ensure that the configuration is deployed and consistent across all environments, the developer can use the following steps:
Create a data patch that contains the configuration for the new website.
Deploy the data patch to all environments.
Use themagento deploy:statuscommand to verify that the configuration has been deployed to all environments.

QUESTION 16
An Adobe Commerce developer was asked to provide additional information on a quote. When getting several quotes, the extension attributes are returned, however, when getting a single quote it fails to be returned.
What is one reason the extension attributes are missing?
A. The developer neglected to add coiiection='trueM to their attribute in etc/extension_attributes.xmi file. O ottribute code='my_attributesM type='MyVendor\MyModule\Api\Data\^AttributeInterface[]M collection='true' />
B. The developer neglected to provide a plugin On Hagento\Quote\Api\CartRepositoryInterface: :get.
C. The developer neglected to implement an observer on the coiiection_ioad_after event.

Correct Answer: A
Section:
Explanation:
The extension attributes are missing because the collection='true' attribute is not set in the etc/extension_attributes.xmi file. This attribute tells Magento that the extension attributes should be returned when the quote is retrieved.
To fix this issue, the developer needs to add the collection='true' attribute to the my_attributes extension attribute.
Once the collection='true' attribute is set, the extension attributes will be returned when the quote is retrieved.

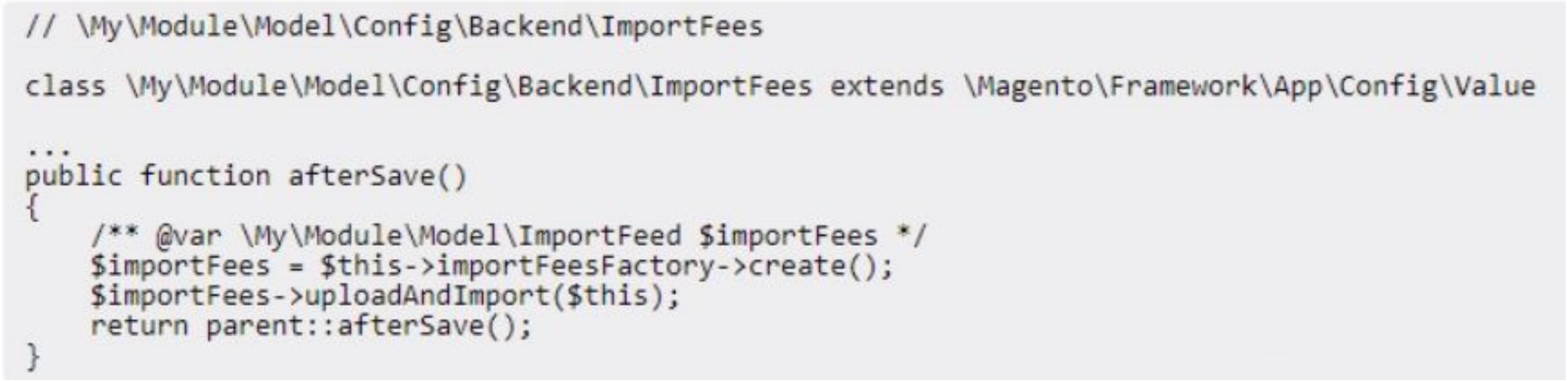
QUESTION 17
A logistics company with an Adobe Commerce extension sends a list of reviewed shipment fees to all its clients every month in a CSV file. The merchant then uploads this CSV file to a 'file upload' field in admin configuration of Adobe Commerce.
What are the two requirements to display the 'file upload' field and process the actual CSV import? (Choose two.)
A)



Add a custom backend model which extends `\Magento\Framework\App\Config\Value` and call `afterSave`:

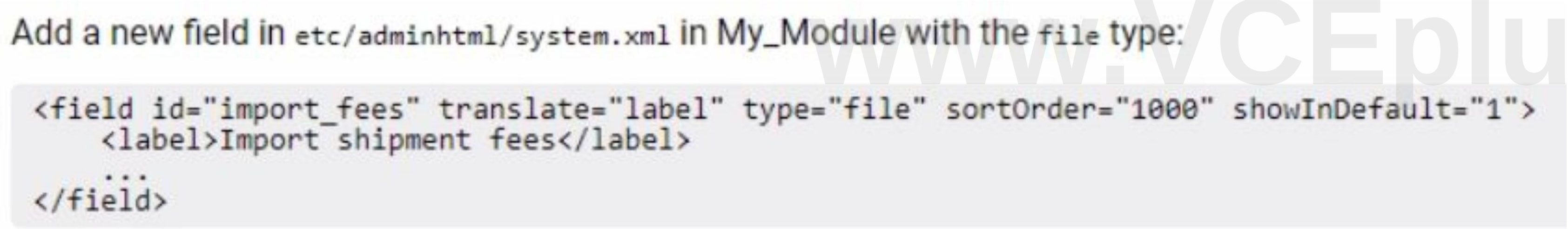
```
// etc/adminhtml/system.xml
<field id="import_fees" ...>
    <label>Import shipment fees</label>
    <backend_model>My\Module\Model\Config\Backend\ImportFees</backend_model>

    ...

</field>
```

B)



```
// \My\Module\Model\Config\Backend\ImportFees

class \My\Module\Model\Config\Backend\ImportFees extends \Magento\Framework\App\Config\Value

...
public function afterSave()
{
    /** @var \My\Module\Model\ImportFeed $importFees */
    $importFees = $this->importFeesFactory->create();
    $importFees->uploadAndImport($this);
    return parent::afterSave();
}
```

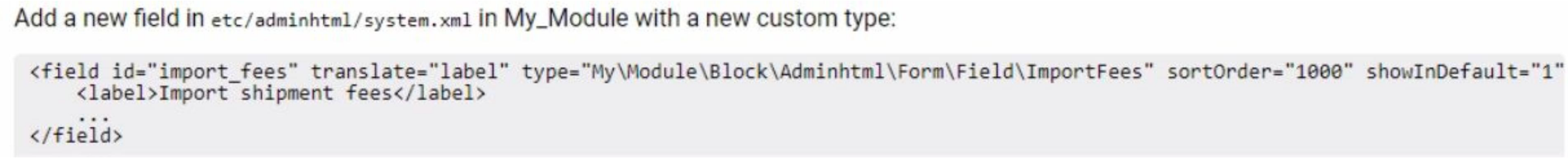
C)



Add a new field in `etc/adminhtml/system.xml` in My_Module with the `file` type:

```
<field id="import_fees" translate="label" type="file" sortOrder="1000" showInDefault="1">
    <label>Import shipment fees</label>
    ...
</field>
```

D)



Add a new field in `etc/adminhtml/system.xml` in My_Module with a new custom type:

```
<field id="import_fees" translate="label" type="My\Module\Block\Adminhtml\Form\Field\ImportFees" sortOrder="1000" showInDefault="1"
    <label>Import shipment fees</label>
    ...
</field>
```

A. Option A
B. Option B
C. Option C
D. Option D

Correct Answer: A, B
Section:
Explanation:
To display the 'file upload' field and process the actual CSV import, the following two requirements must be met:
The developer must create a new system configuration setting that specifies the path to the CSV file.
The developer must create a new controller action that handles the file upload and import process.
The system.xml file is used to define system configuration settings. The following XML snippet shows how to define a new system configuration setting for the CSV file path:

```
XML
<config>
<system>
<config>
<shipment_fees_csv_path>/path/to/csv/file</shipment_fees_csv_path>
</config>
</system>
</config>
```

The Controller\Adminhtml\ShipmentFees controller class is used to handle the file upload and import process. The following code shows how to create a new controller action that handles the file upload and import process:

```
PHP
public function uploadAction()
{
$file = $this->getRequest()->getFile('shipment_fees_csv_file');
if ($file->isUploaded()) {
$importer = new ShipmentFeesImporter();
$importer->import($file);
}
return $this->redirect('adminhtml/system_config/edit/section/shipment_fees');
}
```

QUESTION 18

An Adobe Commerce developer is tasked with adding custom data to orders fetched from the API. While keeping best practices in mind, how would the developer achieve this?

A. Create an extension attribute on Nagento\sales\Api\E)ata\orderinterface and an after plugin on Magento\Sales\Model\Order: :getExtensionAttributes() to add the custom data.

B. Create an extension attribute On Magento\Sales\Api\Data\OrderInterface and an after plugin On Magento\Sales\Api\OrderRepositoryInterface On geto and getListo to add the custom data.

C. Create a before plugin on Magento\sales\model\ResourceModel\order\collection: :load and alter the query to fetch the additional data. Data will then be automatically added to the items fetched from the API.

Correct Answer: B
Section:
Explanation:

The developer should create an extension attribute on the Magento\Sales\Api\Data\OrderInterface interface and an after plugin on the Magento\Sales\Api\OrderRepositoryInterface::get() and Magento\Sales\Api\OrderRepositoryInterface::getList() methods.

The extension attribute will store the custom data. The after plugin will be used to add the custom data to the order object when it is fetched from the API.

Here is the code for the extension attribute and after plugin:

```
PHP
namespace MyVendor\MyModule\Api\Data;
interface OrderExtensionInterface extends \Magento\Sales\Api\Data\OrderInterface
{
/**
* Get custom data.
*
* @return string|null
*/
public function getCustomData();
/**
* Set custom data.
*
* @param string $customData
* @return $this
*/
public function setCustomData($customData);
}
namespace MyVendor\MyModule\Model;
class OrderRepository extends \Magento\Sales\Api\OrderRepositoryInterface
{
/**
* After get order.
*
* @param \Magento\Sales\Api\OrderRepositoryInterface $subject
* @param \Magento\Sales\Api\Data\OrderInterface $order
* @return \Magento\Sales\Api\Data\OrderInterface
*/
public function afterGetOrder($subject, $order)
{
if ($order instanceof OrderExtensionInterface) {
$order->setCustomData('This is custom data');
}
return $order;
}
/**
* After get list.
*
* @param \Magento\Sales\Api\OrderRepositoryInterface $subject
* @param \Magento\Sales\Api\Data\OrderInterface[] $orders
* @return \Magento\Sales\Api\Data\OrderInterface[]
*/
public function afterGetList($subject, $orders)
{
foreach ($orders as $order) {
if ($order instanceof OrderExtensionInterface) {
$order->setCustomData('This is custom data');
}
}
return $orders;
}
}
```

Once the extension attribute and after plugin are created, the custom data will be added to orders fetched from the API.

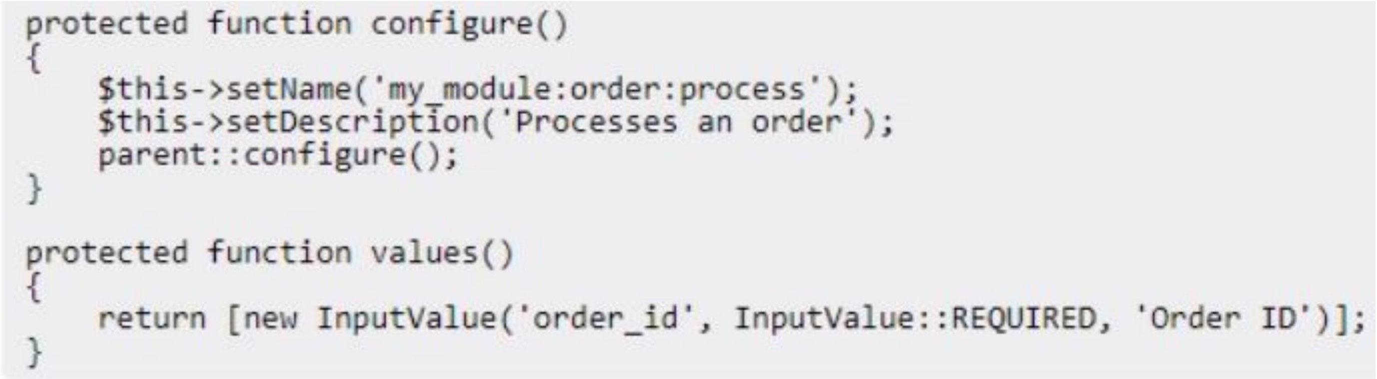
QUESTION 19

An Adobe Commerce developer has created a process that exports a given order to some external accounting system. Launching this process using the Magento CLI with the command php bin/magento my_module:order: process --order_id=<order_id> is required.

Example: php bin/magento my_module:order:process --order_id=1245.

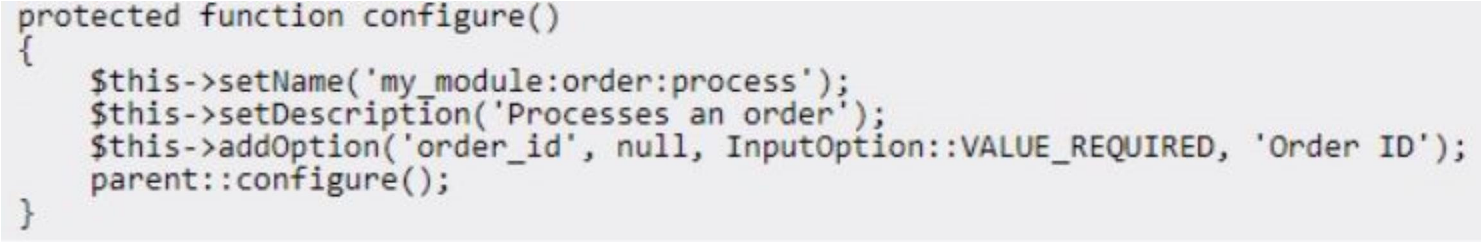
What is the correct way to configure the command?

A)



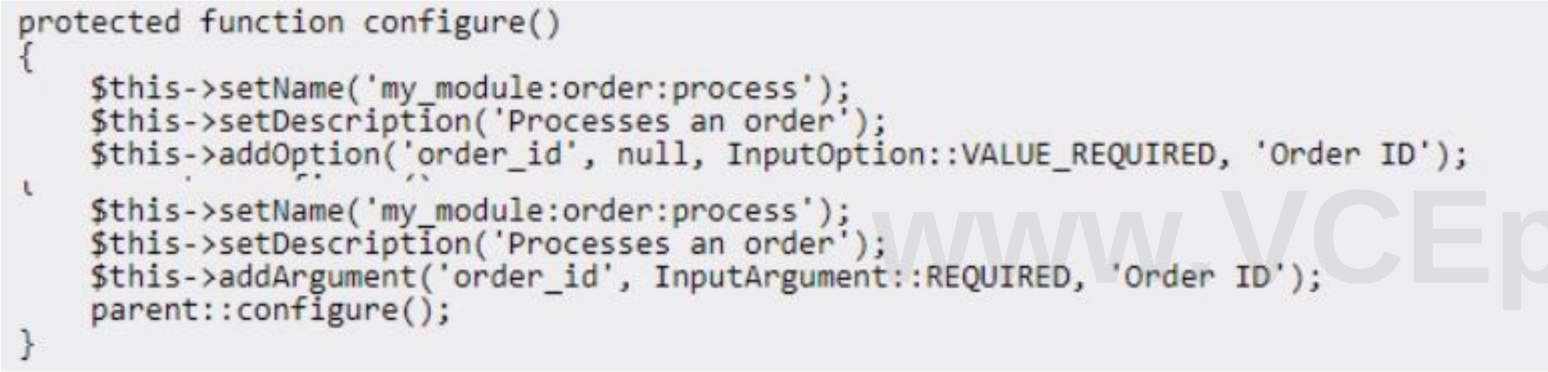
```
protected function configure()
{
    $this->setName('my_module:order:process');
    $this->setDescription('Processes an order');
    parent::configure();
}

protected function values()
{
    return [new InputValue('order_id', InputValue::REQUIRED, 'Order ID')];
}
```

B)



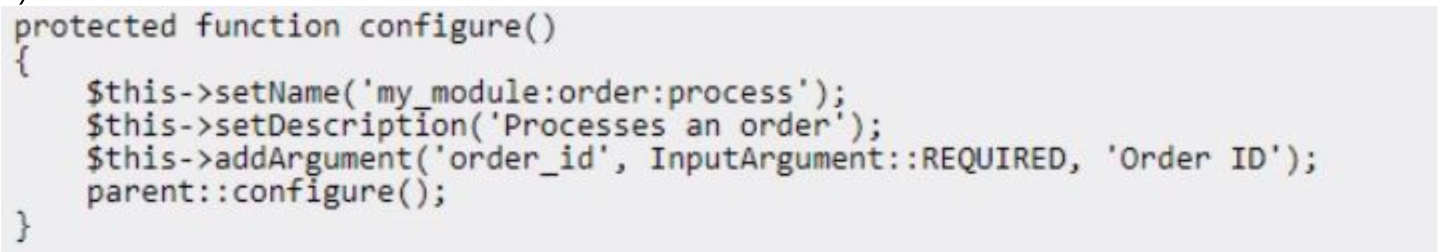
```
protected function configure()
{
    $this->setName('my_module:order:process');
    $this->setDescription('Processes an order');
    $this->addOption('order_id', null, InputOption::VALUE_REQUIRED, 'Order ID');
    parent::configure();
}
```

C)



```
protected function configure()
{
    $this->setName('my_module:order:process');
    $this->setDescription('Processes an order');
    $this->addOption('order_id', null, InputOption::VALUE_REQUIRED, 'Order ID');

    $this->setName('my_module:order:process');
    $this->setDescription('Processes an order');
    $this->addArgument('order_id', InputArgument::REQUIRED, 'Order ID');
    parent::configure();
}
```

D)



```
protected function configure()
{
    $this->setName('my_module:order:process');
    $this->setDescription('Processes an order');
    $this->addArgument('order_id', InputArgument::REQUIRED, 'Order ID');
    parent::configure();
}
```

A. Option B
B. Option C
C. Option C
D. Option D

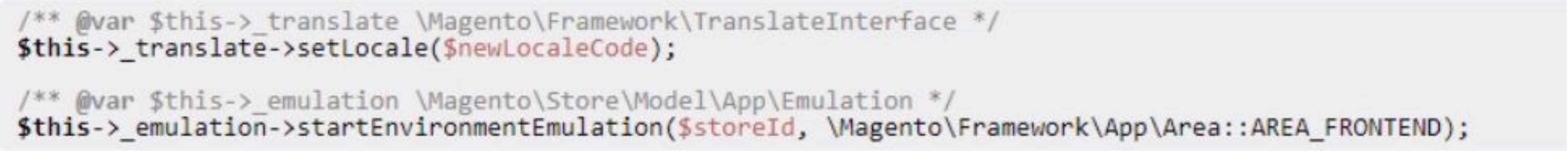
Correct Answer: C
Section:
Explanation:

According to the How to use the Magento command-line interface (CLI) guide, to configure and run the Magento CLI, the developer needs to make the bin/magento file executable and then use it to run commands. To create a custom command, the developer needs to create a class that implements \Symfony\Component\Console\Command\Command and define its name, description, arguments, options, and logic. The developer also needs to register the command in the di.xml file of their module using the Magento\Framework\Console\CommandList argument. Therefore, option C is the correct answer, as it shows the correct class and di.xml code to configure the custom command. Verified

Reference: https://www.a2hosting.com/kb/installable-applications/optimization-and-configuration/magento1/using-the-magento-command-line-interface-cli/

QUESTION 20

An Adobe Commerce developer is working on a Magento 2 instance which contains a B2C and a B2B website, each of which contains 3 different store views for English, Welsh, and French language users. The developer is tasked with adding a link between the B2C and B2B websites using a generic link template which is used throughout the sites, but wants these links to display in English regardless of the store view.
The developer creates a custom block for use with this template, before rendering sets the translate locale and begins environment emulation using the following code:



```
/** @var $this->_translate \Magento\Framework\TranslateInterface */
$this->_translate->setLocale($newLocaleCode);

/** @var $this->_emulation \Magento\Store\Model\App\Emulation */
$this->_emulation->startEnvironmentEmulation($storeId, \Magento\Framework\App\Area::AREA_FRONTEND);
```

They find that the template text is still being translated into each stores language. Why does this occur?

A. startEnvironmffntEmulation() SetS and locks the locale by Using the setLocale() Optional Second $lock parameter, i.e. setLocale($newLocaleCode, true), to override and lock the locale of the emulated store. If this is set and locked initially then the environment emulation will not be able to override this.

B. startEnvironmentEmuiation() resets the translation locale to the one of the emulated stores, which overrides the locale the developer has set when the order of setLocate and startEnvironmentEmulation is used as displayed above.

C. setLocate() does not change translation locale after it has been initially set, the $this->_translate->emulate($newLocaiecode) method exists to temporarily modify this by pushing the new locale to the top of the current emuiatedLocales stack.

Correct Answer: B
Section:
Explanation:

The startEnvironmentEmulation() method resets the translation locale to the one of the emulated stores, which overrides the locale the developer has set when the order of setLocale() and startEnvironmentEmulation() is used as displayed above.
The correct way to achieve the desired result is to use the emulate() method to temporarily modify the translation locale. The following code shows how to do this:
PHP
$this->_translate->emulate('en_US');
// Render the template
$this->_translate->revert();
This code will set the translation locale to English before rendering the template, and then revert the locale back to the default value after the template has been rendered.
The startEnvironmentEmulation() method is used to emulate a different store view or website. This can be useful for testing purposes, or for developing features that need to work in different environments.
The emulate() method is used to temporarily modify the translation locale. This can be useful for rendering templates in a specific language, or for testing features that need to work in different languages.

QUESTION 21

An Adobe Commerce Developer wishes to add an action to a pre-existing route, but does not wish to interfere with the functionality of the actions from the original route.
What must the developer do to ensure that their action works without any side effects in the original module?

A. In the route declaration, use the before or after parameters to load their module in before or after the original module.

B. Inject the new action into the standard router constructor's $actiomist parameter.

C. Add the action into to the controllers/front_name/ in My.Module, Magento will automatically detect and use it.

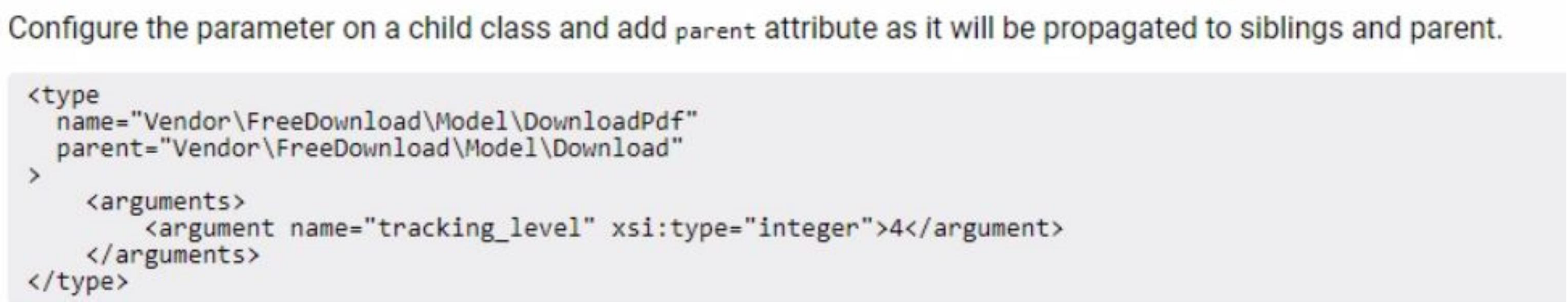
Correct Answer: A
Section:
Explanation:

To add an action to a pre-existing route without interfering with the functionality of the original route, the developer must use the before or after parameters in the route declaration. This will load the developer's module in before or after the original module, respectively.
For example, the following code would add an action to the my_module/index route before the action from the original module:
<route id='my_module/index'>
<before>my_module_before</before>
</route>
The my_module_before action would be executed before the MyModule\Controller\Index action, which would allow the developer to perform any necessary setup before the original action is executed.

QUESTION 22

An Adobe Commerce developer is asked to change the tracking level on a custom module for free downloading of pdf and images.
The module contains following models:
Download class has a parameter for tracking_level.
How will the developer configure the tracking_level parameter, in di.xml.to have a value of 4 for Download class and all classes that extend Download?

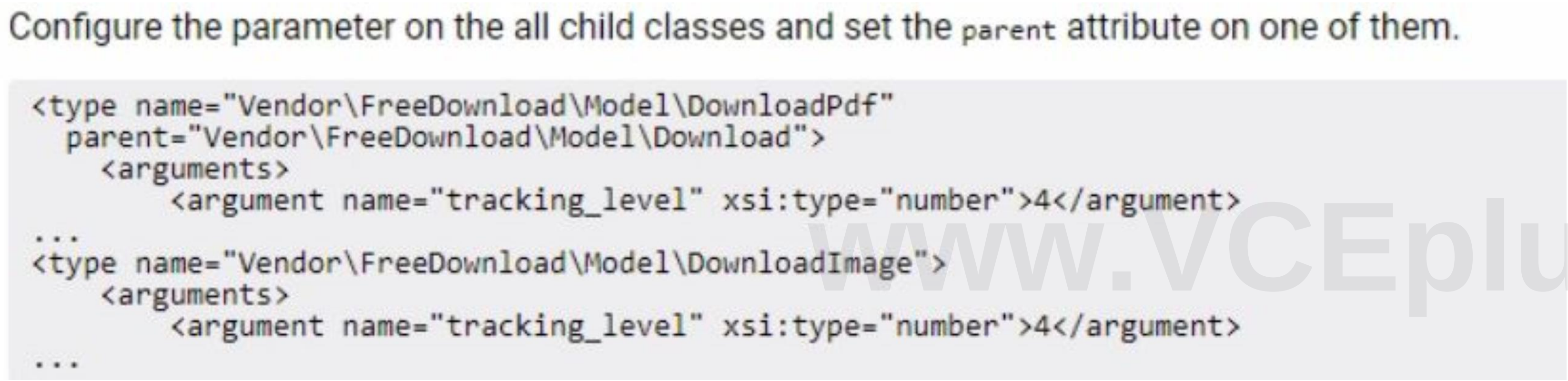
A)



Configure the parameter on a child class and add `parent` attribute as it will be propagated to siblings and parent.

```
<type
  name="Vendor\FreeDownload\Model\DownloadPdf"
  parent="Vendor\FreeDownload\Model\Download"
>
    <arguments>
        <argument name="tracking_level" xsi:type="integer">4</argument>
    </arguments>
</type>
```

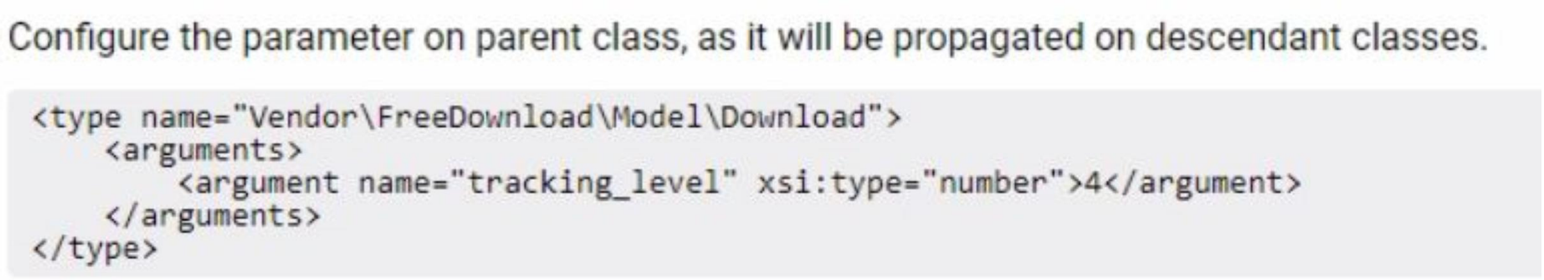
B)



Configure the parameter on the all child classes and set the `parent` attribute on one of them.

```
<type name="Vendor\FreeDownload\Model\DownloadPdf"
  parent="Vendor\FreeDownload\Model\Download">
    <arguments>
        <argument name="tracking_level" xsi:type="number">4</argument>
...
<type name="Vendor\FreeDownload\Model\DownloadImage">
    <arguments>
        <argument name="tracking_level" xsi:type="number">4</argument>
...
```

C)



Configure the parameter on parent class, as it will be propagated on descendant classes.

```
<type name="Vendor\FreeDownload\Model\Download">
    <arguments>
        <argument name="tracking_level" xsi:type="number">4</argument>
    </arguments>
</type>
```

A. Option A
B. Option B
C. Option C

Correct Answer: B
Section:
Explanation:
To configure the tracking_level parameter in di.xml to have a value of 4 for the Download class and all classes that extend Download, the developer would use the following code:
<config>
<global>
<models>
<Vendor\FreeDownload\Model\Download>

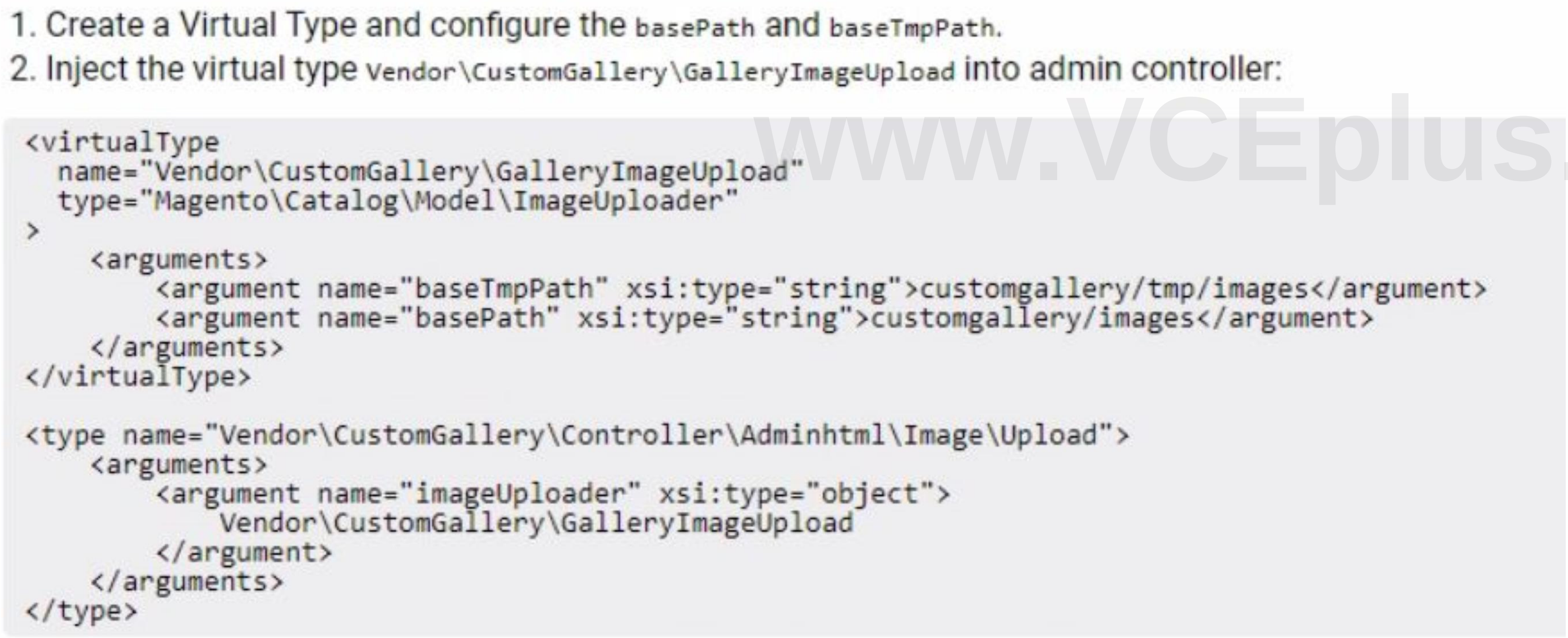
```
<setting name='tracking_level' value='4'/>
</Vendor\FreeDownload\Model\Download>
<Vendor\FreeDownload\Model\DownloadPdf>
<rewrite name='tracking_level' value='4'/>
</Vendor\FreeDownload\Model\DownloadPdf>
<Vendor\FreeDownload\Model\DownloadImage>
<rewrite name='tracking_level' value='4'/>
</Vendor\FreeDownload\Model\DownloadImage>
</models>
</global>
</config>
```

The setting element is used to set a configuration value for a specific model. The rewrite element is used to override the default configuration value for a specific model. In this case, the tracking_level parameter is set to 4 for all models that extend Download.

QUESTION 23

An Adobe Commerce developer is working on a custom gallery extension.
The module uses the Magento\catalog\Model\iinageUploader class for image uploading. The admin controller for custom image uploads is Vendor\CustomGallery\Controller\Adminhtml\Image\Upload.
The images need to be stored in different basePath and baseTmpPath than the default ones.
How can the default imageuploader class be extended and used without affecting the other modules that are already using it?

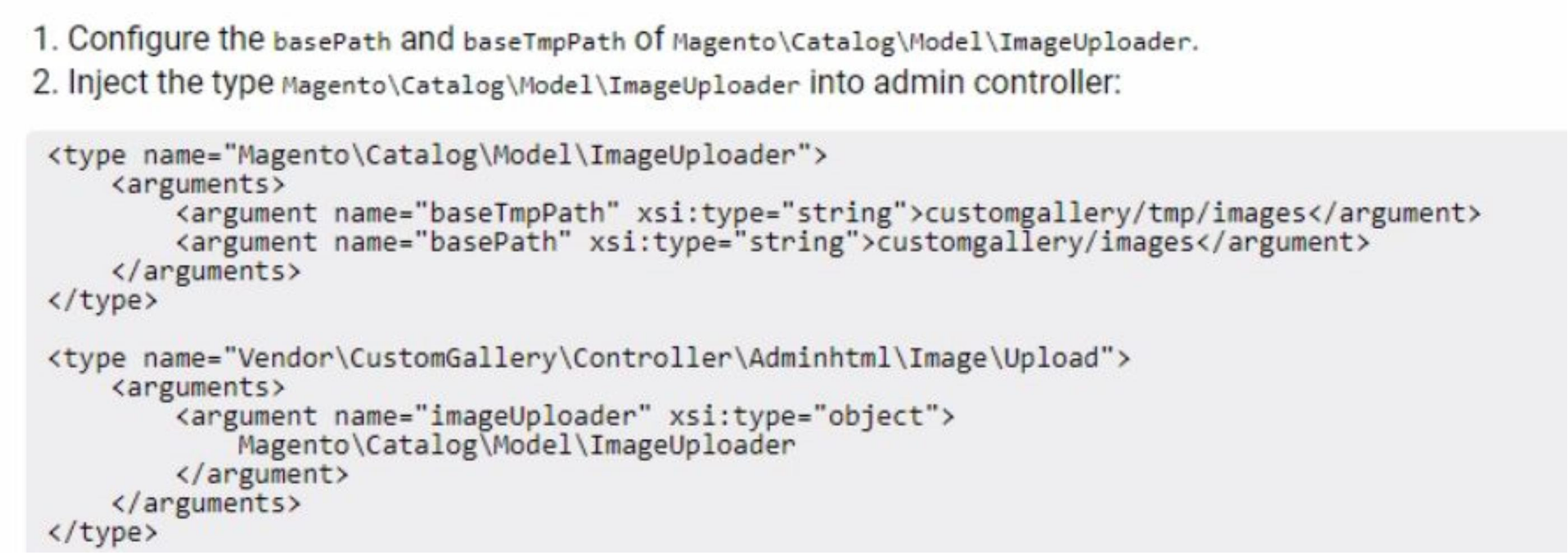
A)



1. Create a Virtual Type and configure the `basePath` and `baseTmpPath`.
2. Inject the virtual type `Vendor\CustomGallery\GalleryImageUpload` into admin controller:

```
<virtualType
  name="Vendor\CustomGallery\GalleryImageUpload"
  type="Magento\Catalog\Model\ImageUploader"
>
    <arguments>
        <argument name="baseTmpPath" xsi:type="string">customgallery/tmp/images</argument>
        <argument name="basePath" xsi:type="string">customgallery/images</argument>
    </arguments>
</virtualType>

<type name="Vendor\CustomGallery\Controller\Adminhtml\Image\Upload">
    <arguments>
        <argument name="imageUploader" xsi:type="object">
            Vendor\CustomGallery\GalleryImageUpload
        </argument>
    </arguments>
</type>
```

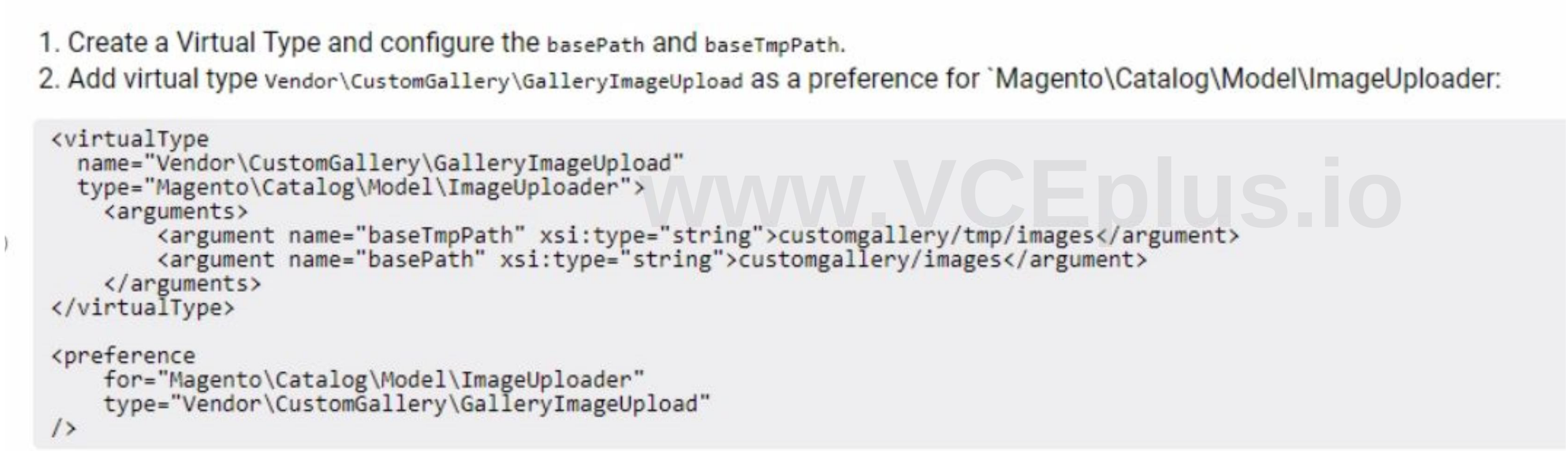
B)



1. Configure the `basePath` and `baseTmpPath` of `Magento\Catalog\Model\ImageUploader`.
2. Inject the type `Magento\Catalog\Model\ImageUploader` into admin controller:

```
<type name="Magento\Catalog\Model\ImageUploader">
    <arguments>
        <argument name="baseTmpPath" xsi:type="string">customgallery/tmp/images</argument>
        <argument name="basePath" xsi:type="string">customgallery/images</argument>
    </arguments>
</type>

<type name="Vendor\CustomGallery\Controller\Adminhtml\Image\Upload">
    <arguments>
        <argument name="imageUploader" xsi:type="object">
            Magento\Catalog\Model\ImageUploader
        </argument>
    </arguments>
</type>
```

C)



1. Create a Virtual Type and configure the `basePath` and `baseTmpPath`.
2. Add virtual type `Vendor\CustomGallery\GalleryImageUpload` as a preference for `Magento\Catalog\Model\ImageUploader:

```
<virtualType
  name="Vendor\CustomGallery\GalleryImageUpload"
  type="Magento\Catalog\Model\ImageUploader">
    <arguments>
        <argument name="baseTmpPath" xsi:type="string">customgallery/tmp/images</argument>
        <argument name="basePath" xsi:type="string">customgallery/images</argument>
    </arguments>
</virtualType>

<preference
    for="Magento\Catalog\Model\ImageUploader"
    type="Vendor\CustomGallery\GalleryImageUpload"
/>
```

A. Option A
B. Option B
C. Option C

Correct Answer: B
Section:
Explanation:
According to the ImageUploader component guide for Magento 2 developers, the ImageUploader UI component gives users the ability to upload images to the Magento Media Gallery. This component is a variation of the FileUploader component and uses the same configuration settings. The ImageUploader component uses the Magento\catalog\Model\iinageUploader class for image uploading, which has properties such as basePath and baseTmpPath that define where the images are stored. To extend the default imageuploader class and use it without affecting the other modules that are already using it, the developer needs to create a virtual type of this class in their module's di.xml file and specify different values for basePath and baseTmpPath. The developer also needs to inject their virtual type into their admin controller using the imageUploader argument. Therefore, option B is the correct answer, as it shows the correct di.xml and controller code to extend and use the imageuploader class. Verified
Reference: https://devdocs.magento.com/guides/v2.3/ui_comp_guide/components/image-uploader/

QUESTION 24

An Adobe Commerce developer has added a new configuration field to the admin are
A. The path for this option is general/store_information/out_of_hours_phone.
Keeping simplicity in mind, how would the developer ensure this option contains a valid US telephone number?
B. Add <validate>phoneUS</validate> to the field in system.xml.
C. Create a backend model to check the validity of the phone number entered.
D. Add <validate type='phoneUS'/> to the field in system.xml.

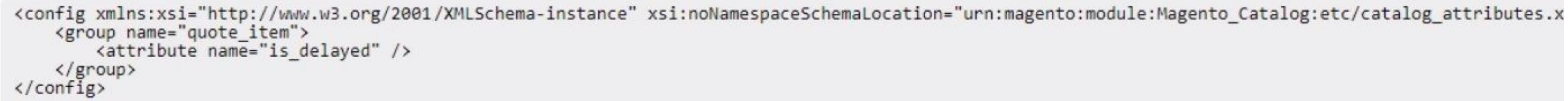
Correct Answer: A
Section:
Explanation:
According to the Magento Stack Exchange answer, system.xml is a file that defines the configuration fields for the admin area. Each field can have a validate attribute that specifies a validation rule for the field value. Magento provides some built-in validation rules, such as phoneUS, which validates a US telephone number. Therefore, to ensure that the option contains a valid US telephone number, the developer needs to add <validate>phoneUS</validate> to the field in system.xml. Verified
Reference: https://magento.stackexchange.com/questions/104570/magento-2-system-xml-validation-rules

QUESTION 25
An Adobe Commerce developer is tasked with creating a custom block that will be displayed on every page in the footer of the site.
After completing and optimizing the development, the developer notices that the block takes too much time to be generated on each page and decides to store it in the system cache after enabling it for all cache groups.
What would be the minimum requirement to achieve this?
A. Set a value for the cache_Lifetime data property of the block.
B. Set a value for cache_key data property of the block.
C. Set values for both cache_lifetime and cache_key data properties of the block.

Correct Answer: C
Section:
Explanation:

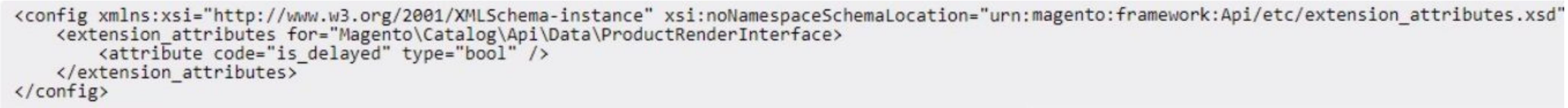
To store a block in the system cache, the developer needs to set values for both the cache_lifetime and cache_key data properties of the block. The cache_lifetime property specifies how long the block should be cached, and the cache_key property specifies a unique identifier for the block.
The following code shows how to set the cache_lifetime and cache_key data properties of a block:
PHP
$block->setData('cache_lifetime', 600);
$block->setData('cache_key', 'my_custom_block');
Once the cache_lifetime and cache_key data properties have been set, the block will be stored in the system cache and will not be regenerated on each page load.

QUESTION 26
There is the task to create a custom product attribute that controls the display of a message below the product title on the cart page, in order to identify products that might be delivered late.
The new EAV attribute is_delayed has been created as a boolean and is working correctly in the admin panel and product page.
What would be the next implementation to allow the is_delayed EAV attribute to be used in the .phtml cart page such as $block->getProduct()->getIsDelayed()?
A)
Create a new file etc/catalog_attributes.xmi:



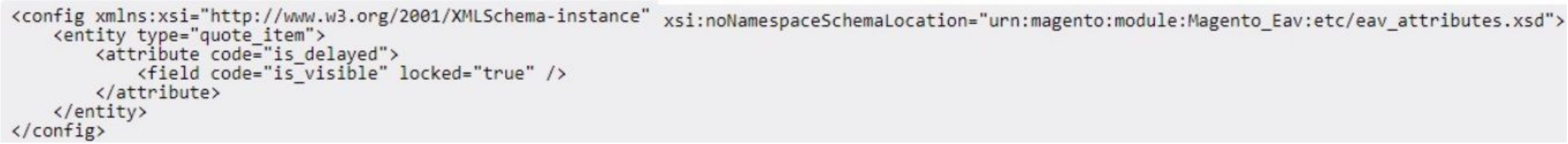
```
<config xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="urn:magento:module:Magento_Catalog:etc/catalog_attributes.x
    <group name="quote_item">
        <attribute name="is_delayed" />
    </group>
</config>
```

B)
Create a new file etc/extension attributes.xmi:



```
<config xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="urn:magento:framework:Api/etc/extension_attributes.xsd"
    <extension_attributes for="Magento\Catalog\Api\Data\ProductRenderInterface>
        <attribute code="is_delayed" type="bool" />
    </extension_attributes>
</config>
```

C)
Create a new file etc/eav attributes.xmi:



```
<config xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="urn:magento:module:Magento_Eav:etc/eav_attributes.xsd">
    <entity type="quote_item">
        <attribute code="is_delayed">
            <field code="is_visible" locked="true" />
        </attribute>
    </entity>
</config>
```

A. Option A
B. Option B
C. Option C

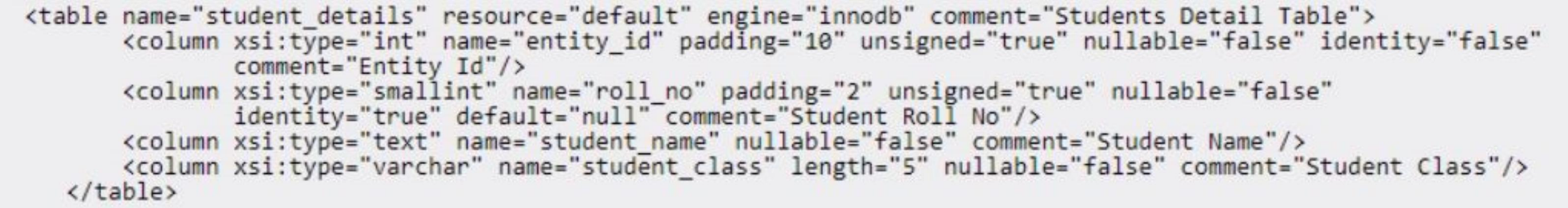
Correct Answer: A
Section:
Explanation:
To allow the is_delayed EAV attribute to be used in the .phtml cart page, the developer needs to create a new file called etc/catalog_attributes.xmi. This file will contain the definition of the is_delayed attribute.
The following code shows how to create the etc/catalog_attributes.xmi file:
XML
<?xml version='1.0'?>
<catalog_attributes>
<label>Is Delayed</label>
<note>This attribute indicates whether the product is delayed.</note>
<sort_order>10</sort_order>
<required>false</required>
</catalog_attributes>
Once the etc/catalog_attributes.xmi file has been created, the is_delayed attribute will be available in the .phtml cart page. The attribute can be accessed using the getIsDelayed() method of the Product class.
PHP
$product = $block->getProduct();
$isDelayed = $product->getIsDelayed();
The isDelayed variable will contain the value of the is_delayed attribute. If the value of the attribute is 1, then the product is delayed. If the value of the attribute is 0, then the product is not delayed.

QUESTION 27
An Adobe Commerce developer is creating a module (Vendor.ModuleName) to be sold on the Marketplace. The new module creates a database table using declarative schema and now the developer needs to make sure the table is removed when the module is disabled.
What must the developer do to accomplish this?
A. There is nothing further the developer needs to do. The table will be removed when the module is disabled and bin/magento setup:upgrade is run.
B. There is nothing further the developer needs to do. The table will be removed when the when bin/magento module:uninstall vendor_ModuleName is run.
C. Add a schema patch that implements Magento\Framework\setup\Patch\PatchRevertabieinterface and drops the table in the revert function.

Correct Answer: C
Section:
Explanation:
According to the Declarative Schema Overview guide for Magento 2 developers, declarative schema is a new feature that allows developers to declare the final desired state of the database and has the system adjust to it automatically, without performing redundant operations. However, declarative schema does not support uninstalling modules or reverting changes. To remove a table when a module is disabled, the developer

needs to add a schema patch that implements Magento\Framework\setup\Patch\PatchRevertabieinterface and drops the table in the revert function. The revert function will be executed when the module is disabled using bin/magento module:disable command. Verified
Reference: https://devdocs.magento.com/guides/v2.3/extension-dev-guide/declarative-schema/

QUESTION 28
An Adobe Commerce developer is trying to create a custom table using declarative schema, but is unable to do so.



```
  <table name="student_details" resource="default" engine="innodb" comment="Students Detail Table">
         <column xsi:type="int" name="entity_id" padding="10" unsigned="true" nullable="false" identity="false"
                 comment="Entity Id"/>
         <column xsi:type="smallint" name="roll_no" padding="2" unsigned="true" nullable="false"
                 identity="true" default="null" comment="Student Roll No"/>
         <column xsi:type="text" name="student_name" nullable="false" comment="Student Name"/>
         <column xsi:type="varchar" name="student_class" length="5" nullable="false" comment="Student Class"/>
     </table>
```

What are two errors in the snippet above? (Choose two.)
A. Column (roll_no) does not have index. It is needed since attribute identity is set to true.
B. Column (entity_id) does not have index. It is needed since attribute identity is set to false.
C. Column (student_name) does not have attribute length.
D. null is not a valid value for column (roll_no).

Correct Answer: A, C
Section:
Explanation:
The correct answers are A and C.
The errors in the snippet are:
Columnroll_nodoes not have an index. It is needed sinceattribute_identityis set totrue.
Columnstudent_namedoes not have an attribute length.
The attribute_identity attribute specifies whether the primary key of the table should be auto-incremented. If attribute_identity is set to true, then the roll_no column must have an index. The student_name column does not have an attribute length, which is required for string columns.
The following code shows how to fix the errors:
XML
<table name='vendor_module_table'>
<entity_id>
<type>int</type>
<identity>true</identity>
<unsigned>true</unsigned>
<nullable>false</nullable>
</entity_id>
<roll_no>
<type>int</type>
<identity>false</identity>
<unsigned>true</unsigned>
<nullable>false</nullable>
true
<index>true</index>
</roll_no>
<student_name>
<type>string</type>
<length>255</length>

<nullable>false</nullable>
</student_name>
</table>
Once the errors have been fixed, the table can be created successfully.

QUESTION 29

An Adobe Commerce developer wants to create a product EAV attribute programmatically which should appear as WYSIWYG in the admin panel. They have made sure that wysiwyg_enabled has been set to true, however, the attribute is not appearing as WYSIWYG in the admin panel.
What would be a possible reason?

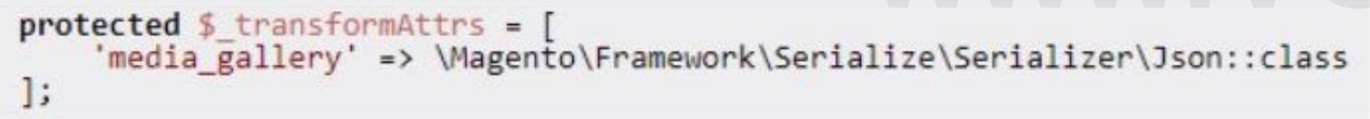
A. The is_html_allowed_on_front Option iS Set tO false.
B. The input type is not set to text.
C. The input type is not set to textarea.

Correct Answer: C
Section:
Explanation:
The input_type attribute of a product EAV attribute specifies the type of input field that will be used to enter the value of the attribute in the admin panel. The textarea input type is used for WYSIWYG fields. If the input_type attribute is not set to textarea, then the attribute will not appear as WYSIWYG in the admin panel.
To fix this, the developer should set the input_type attribute to textarea.

QUESTION 30

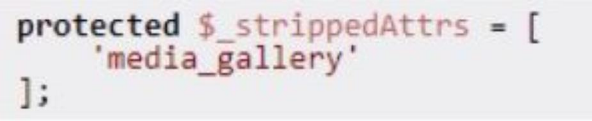
An Adobe Commerce Developer has written an importer and exporter for a custom entity. The client is using this to modify the exported data and then re-importing the file to batch update the entities.
There is a text attribute, which contains information related to imagery in JSON form, media_gallery. This is not a field that the client wants to change, but the software they are using to edit the exported data seems to be modifying it and not allowing it to import correctly.
How would the developer prevent this?

A) Specify a serializer class for the attribute using the $_transformAttrs class property array for both the exporter and importer so it gets converted:



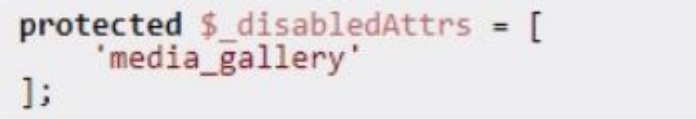
```
protected $_transformAttrs = [
    'media_gallery' => \Magento\Framework\Serialize\Serializer\Json::class
];
```

B) Strip the attribute from the imported file by adding it to the s_strippedAttrs class property array:



```
protected $_strippedAttrs = [
    'media_gallery'
];
```

C) Prevent it from being exported by adding it to the $_disat>iedAttrs class property array:



```
protected $_disabledAttrs = [
    'media_gallery'
];
```

A. Option A
B. Option B
C. Option C

Correct Answer: A
Section:
Explanation:
The _transformAttrs class property array of the importer and exporter classes can be used to specify a serializer class for a particular attribute. The serializer class will be used to convert the attribute value from one format to another when the data is exported or imported.
In this case, the developer can specify a serializer class that will convert the JSON data in the media_gallery attribute to a string. This will prevent the software that the client is using to modify the exported data from changing the JSON data.
The following code shows how to specify a serializer class for the media_gallery attribute:

```
PHP
class MySerializer
{
public function serialize($value)
{
return json_encode($value);
}
public function deserialize($value)
{
return json_decode($value);
}
}
$importer->setSerializer('media_gallery', MySerializer::class);
$exporter->setSerializer('media_gallery', MySerializer::class);
```

Once the serializer class has been specified, the JSON data in the media_gallery attribute will be converted to a string when the data is exported or imported. This will prevent the software that the client is using to modify the exported data from changing the JSON data.

QUESTION 31

The developer is required to convert a modules database scripts from old install/upgrade setup files to a data patches format and does not want to apply database changes that were already done by install/upgrade scripts.

The current module version is 1.5.4.

What would be the recommended solution to skip changes that were already applied via old format (install/upgrade scripts)?

A. Implement Patchversioninterface and return 1.5.4 on the getversion() method.

B. Inside apply() method, check for module version and run the code if version is less than 1.5.4.

C. This is not possible. A module cannot implement both data patch and install scripts.

Correct Answer: A
Section:
Explanation:

According to the Develop data and schema patches guide for Magento 2 developers, data patches are classes that contain data modification instructions. They are defined in a <Vendor>/<Module_Name>/Setup/Patch/Data/<Patch_Name>.php file and implement MagentoFrameworkSetupPatchDataPatchInterface. Data patches can also implement Patchversioninterface to specify the module version that the patch is associated with. The getVersion() method returns the module version as a string. To skip changes that were already applied via old format (install/upgrade scripts), the developer should implement Patchversioninterface and return 1.5.4 on the getVersion() method. This way, the data patch will only be applied if the module version is greater than or equal to 1.5.4. Verified

Reference: https://devdocs.magento.com/guides/v2.3/extension-dev-guide/declarative-schema/data-patches.html

QUESTION 32

An Adobe Commerce developer has created a module that adds a product attribute to all product types via a Data Patch-According to best practices, how would the developer ensure this product attribute is removed in the event that the module is uninstalled at a later date?

A. Add an Uninstall.php file extending \I1agento\Framework\Setup\UninstallInterface tO the module's Setup directory and implement the uninstall method.

B. Add instructions to the module's README.md file instructing merchants and developers that they must manually remove this attribute if they want to uninstall the module.

C. Make the Data Patch implement \Magento\Framework\setup\Patch\PatchRevertabieinterface and implement the revert method to remove the product attribute.

Correct Answer: C
Section:
Explanation:

According to the Develop data and schema patches guide for Magento 2 developers, data patches can also implement PatchRevertabieinterface to provide rollback functionality for their changes. The revert() method contains the instructions to undo the data modifications made by the patch. To ensure that the product attribute is removed when the module is uninstalled, the developer should make the data patch implement PatchRevertabieinterface and implement the revert method to remove the product attribute using EavSetupFactory or AttributeRepositoryInterface. Verified

Reference: https://devdocs.magento.com/guides/v2.3/extension-dev-guide/declarative-schema/data-patches.html

QUESTION 33

An Adobe Commerce developer successfully added a new column to the customers grid. This column needs the data to be formatted before showing its content in the grid.
According to best practices, how would the developer add the custom logic to render the column?

A. 1- Create an after pluginforMagento\Ui\Component\Listing\Columns\Column::prepareColumn().
2- Add the custom logic within the afterPreparecoiumn method.

B. 1- Create a custom class extending flagento\Ui\Component\Listing\Columns\Colunm.
2- Add the custom logic within the prepareDataSource method.
3- Add an attribute class to the column node within the module's customer_listing.xml.

C.
1- Override the Magento\Customer\Ui\Component\DataProvider Class using a preference.
2- Override the getData() method and add the custom logic per row.

Correct Answer: A
Section:
Explanation:

QUESTION 34

An integration named Marketing is created on the Adobe Commerce instance. The integration has access on Magento_Customer:: customer resources and the access token is xxxxxx.
How would the rest API be called to search the customers?

A. Using the integration access token as Bearer: curl -X GET https://magentourl/rest/Vl/customers/search?searchCriteria... -H 'Authorization: Bearer XXXXXX'

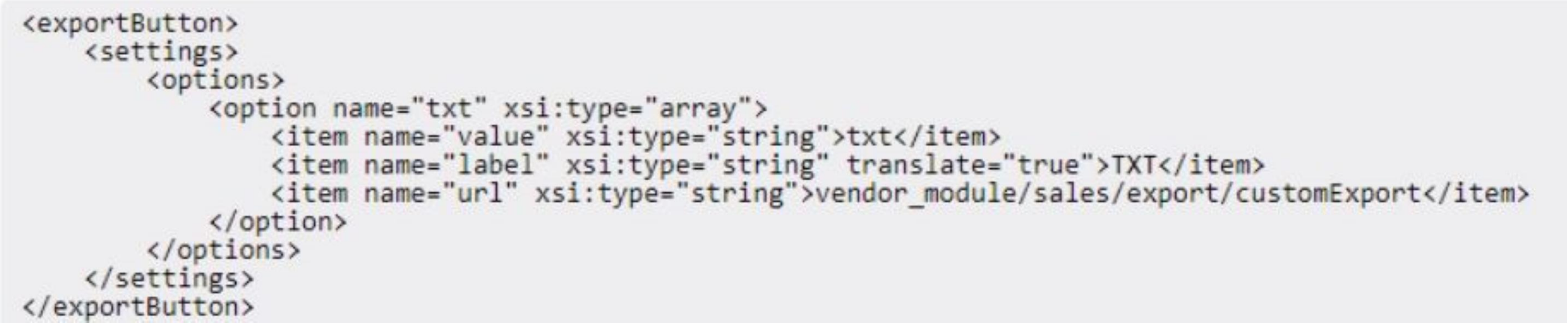
B. Passing integration name and access token as http auth credentials: curl -X GET https ://Marketing:XXXXXX(Slmagentourl/rest/Vl/customers/search?5earchCriteria . . . Using integration name as username and access token as password, get the admin token (yyyyyy) via: curl -X POST https://magentourl/rest/Vl/integration/admin/token -d '{'username':'Marketing', 'password':'XXXXXX'}' -H 'Content-

C. Type: application/json' Use the admin token as Bearer curl -X GET https://magentourl/rest/Vl/customers/search?searchCriteria... -H 'Authorization: Bearer YYYYYY'

Correct Answer: B
Section:
Explanation:
According to the Magento Stack Exchange answer, UI components are used to render various elements on Magento admin pages, such as grids, forms, buttons, etc. UI components are defined in XML files that are located in the view/adminhtml/ui_component directory of each module. To add a custom logic to render a column in a grid, the developer should create a custom class extending MagentoUiComponentListingColumnsColumn and add the custom logic within the prepareDataSource method. This method receives an array of data sources and modifies them according to the column logic. The developer should also add an attribute class to the column node within the module's customer_listing.xml file and specify their custom class name as its value. Verified
Reference: https://magento.stackexchange.com/questions/317821/how-to-add-custom-logic-to-render-a-column-in-a-grid-in-magento-2

QUESTION 35

An Adobe Commerce developer is tasked with adding an new export option for the order grid, they have added the following code for the export button within sales_order_grid.xml:



```
<exportButton>
    <settings>
        <options>
            <option name="txt" xsi:type="array">
                <item name="value" xsi:type="string">txt</item>
                <item name="label" xsi:type="string" translate="true">TXT</item>
                <item name="url" xsi:type="string">vendor_module/sales/export/customExport</item>
            </option>
        </options>
    </settings>
</exportButton>
```

Upon testing, they are getting redirected, what would be a cause for this error?

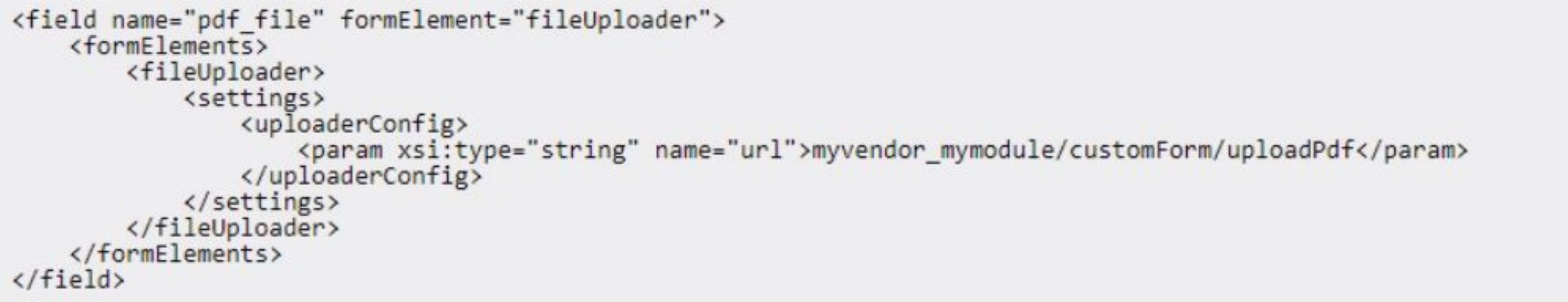
A. The option's uri attribute is not valid.

B. The layout cache needs to be refreshed.

C. The developer has to add a formkey for the new export option.

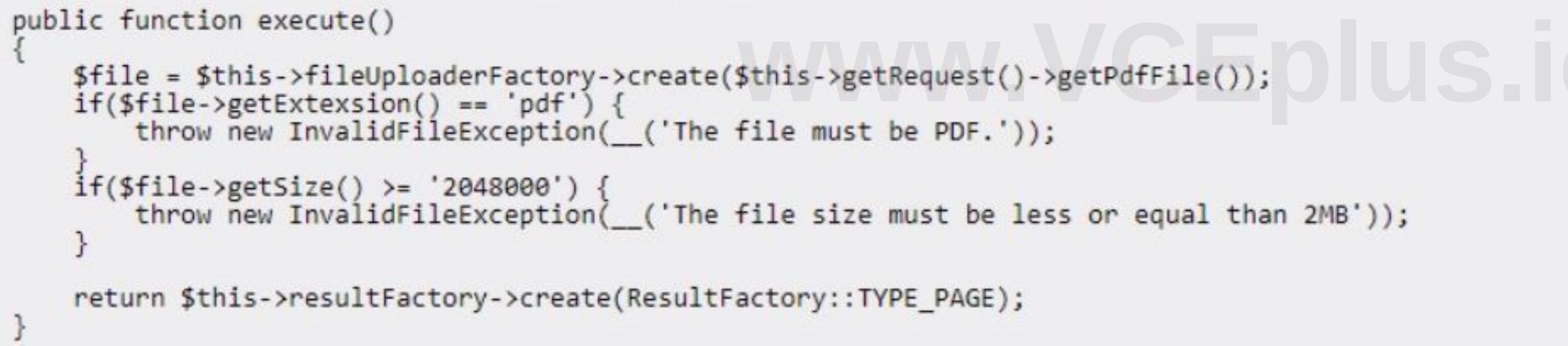
Correct Answer: C
Section:
Explanation:
The developer has to add a formkey for the new export option because the formkey is required for security reasons. Without the formkey, the request will be rejected and redirected to the dashboard page. Verified Reference: [Magento 2.4 User Guide] [Magento 2.4 DevDocs]

QUESTION 36
An Adobe Commerce developer is tasked to add a file field to a custom form in the administration panel, the field must accept only .PDF files with size less or equal than 2 MB. So far the developer has added the following code within the form component xml file, inside the fieldset node:



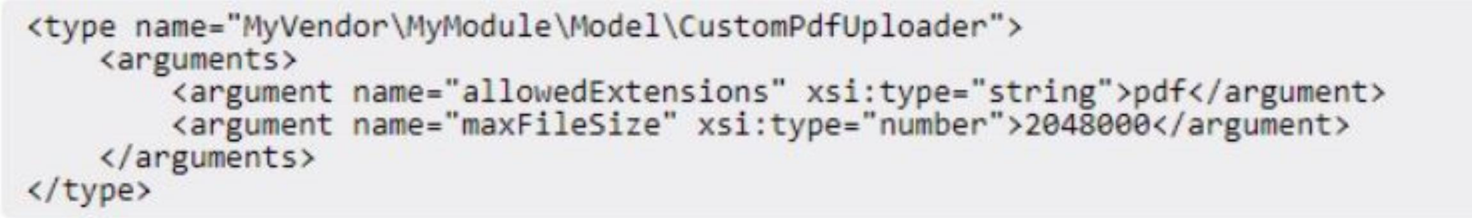
```
<field name="pdf_file" formElement="fileUploader">
    <formElements>
        <fileUploader>
            <settings>
                <uploaderConfig>
                    <param xsi:type="string" name="url">myvendor_mymodule/customForm/uploadPdf</param>
                </uploaderConfig>
            </settings>
        </fileUploader>
    </formElements>
</field>
```

How would the developer implement the validations?
A)
Add the Validations Within the HyVendor\MyModule\Controller\Adminhtml\CustomEntity\UploadPdf Controller



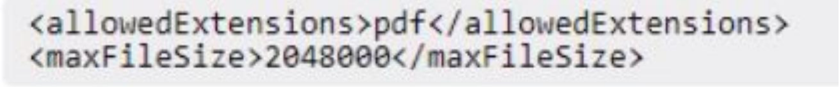
```
public function execute()
{
    $file = $this->fileUploaderFactory->create($this->getRequest()->getPdfFile());
    if($file->getExtesion() == 'pdf') {
        throw new InvalidFileException(__('The file must be PDF.'));
    }
    if($file->getSize() >= '2048000') {
        throw new InvalidFileException(__('The file size must be less or equal than 2MB'));
    }

    return $this->resultFactory->create(ResultFactory::TYPE_PAGE);
}
```

B)
Add a virtual type forMyvendor\MyModuie\Modei\customPdfupioader specifying the aiiowedExtensions and the maxFiiesize for the constructor, within the module's di.xmi:



```
<type name="MyVendor\MyModule\Model\CustomPdfUploader">
    <arguments>
        <argument name="allowedExtensions" xsi:type="string">pdf</argument>
        <argument name="maxFileSize" xsi:type="number">2048000</argument>
    </arguments>
</type>
```

C)
Add the following code inside the<settings> node:



```
<allowedExtensions>pdf</allowedExtensions>
<maxFileSize>2048000</maxFileSize>
```

A. Option A
B. Option B
C. Option C

Correct Answer: B
Section:
Explanation:
The developer can add a virtual type for Myvendor\MyModuie\Modei\customPdfupioader specifying the aiiowedExtensions and the maxFiiesize for the constructor, within the module's di.xmi. This way, the developer can reuse the existing file uploader class and customize it for the specific field without modifying the core code. Verified
Reference: [Magento 2.4 DevDocs] [Magento Stack Exchange]

QUESTION 37
An Adobe Commerce developer wants to generate a list of products using ProductRepositoryInterf ace and search for products using a supplier_id filter for data that is stored in a standalone table (i.e., not in an EAV attribute).
Keeping maintainability in mind, how can the developer add the supplier ID to the search?
A. Write a before plugin on \Hagento\catalogVtodel\ProductRepository: :geti_ist() and register the search criteria passed. Write an event observer to 0 listen for the event cataiog_product_coiiection_ioad_before. Iterate through the registered search criteria, and if found, apply the needed join and filter to the events scollection.
B. Add a CUStOm filter to the Virtual type 'agento\Catalog\Model\Api\SearchCriteria\CollectionProcessor\ProductFilterProce5sor for supplier_id field. In the custom filter, apply the needed join and filter to the passed $collection.
C. Write a before plugin On \Magento\Framework\Api\SearchCriteria\CollectionProcessorInterface: :process(). Iterate through the $searchCriteria provided for supplier_id, and if found, apply the needed join and filter to the passed scollection.

Correct Answer: B
Section:
Explanation:
The developer can add a custom filter to the virtual type Magento\Catalog\Model\Api\SearchCriteria\CollectionProcessor\ProductFilterProce5sor for supplier_id field. In the custom filter, the developer can apply the needed join and filter to the passed $collection. This is the recommended way to extend the search criteria for products using dependency injection and plugins. Verified
Reference: [Magento 2.4 DevDocs] [Magento Stack Exchange]

QUESTION 38
When researching some issues with the indexer, an Adobe Commerce developer is seeing errors in the logs similar to Memory size allocated for the temporary table is more than 20% of innodb_buffer_pool_size. It is suggested that the client update innodb_buf f er_pool_size or decrease the batch size value.
Why does decreasing the batch size value improve performance?
A. This decreases memory usage for the temporary table.
B. This allows for a longer timeout per batch process.
C. This allows for more PHP threads to be utilized during the process.

Correct Answer: A
Section:
Explanation:
Decreasing the batch size value improves performance by reducing the memory usage for the temporary table. The batch size value determines how many rows of data are processed at a time by the indexer. A large batch size value can cause the allocated memory size for the temporary table to exceed 20% of innodb_buffer_pool_size, which can result in errors and slow down the indexing process. By lowering the batch size value, the indexer can process the data more efficiently and avoid memory issues. Verified
Reference: [Magento 2.4 DevDocs] [Magento Stack Exchange]

QUESTION 39
An international merchant is complaining that changes are taking too long to be reflected on the frontend after a full product import.
Thinking it may be database issues, the Adobe Commerce developer collects the following entity counts:
* Categories: 900
* Products: 300k
* Customers: 700k
* Customer groups : 106
* Orders: 1600k

* Invoices: 500k
* Creditmemos: 50k
* Websites : 15
* Stores : 45

What is a probable cause for this?

A. The combination of the number of products, categories and stores is too big. This leads to a huge amount of values being stored in the flat catalog indexes which are too large to be processed at a normal speed.
B. The combination of the number of orders, customers, invoices and creditmemos is too big. This leads to a huge amount of values being stored in the customer grid index which is too large to be processed at a normal speed.
C. The combination of the number of products, customer groups and websites is too big. This leads to a huge amount of values being stored in the price index which is too large to be processed at a normal speed.

Correct Answer: C
Section:
Explanation:

The probable cause for the delay in reflecting the changes on the frontend after a full product import is the combination of the number of products, customer groups and websites. This leads to a huge amount of values being stored in the price index which is too large to be processed at a normal speed. The price index calculates the final price of each product for each customer group and website, taking into account various factors such as tax, discounts, catalog price rules, etc. When there are many products, customer groups and websites, the price index becomes very complex and time-consuming to update. Verified
Reference: [Magento 2.4 DevDocs] [Magento Stack Exchange]

QUESTION 40

When checking the cron logs, an Adobe Commerce developer sees that the following job occurs daily: main.INFO: Cron Dob inventory_cleanup_reservations is successfully finished. However, the inventory_reservation table in the database is not emptied. Why are there records remaining in the inventory_reservation table?

A. Only reservations matching canceled orders are removed by the cron job.
B. Only reservations no longer needed are removed by the cron job.
C. The 'Auto Cleanup' feature from Multi Source Inventory was disabled in configuration.

Correct Answer: B
Section:
Explanation:

The reason why there are records remaining in the inventory_reservation table is that only reservations no longer needed are removed by the cron job. The inventory_reservation table tracks the quantity of each product in each order and creates a reservation for each product when an order is placed, shipped, cancelled or refunded. The initial reservation has a negative quantity value and the subsequent reservations have positive values. When the order is complete, the sum of all reservations for the product is zero. The cron job removes only those reservations that have a zero sum from the table, leaving behind any reservations that are still needed for incomplete orders. Verified
Reference: [Magento 2.4 DevDocs] [Magento Stack Exchange]

QUESTION 41

An Adobe Commerce developer is asked to create a new payment method for their project. This project has administrators who use the backend to manage customer information and occasionally place orders. When testing the new payment method on the frontend everything worked as expected, however, the payment method is missing in the admin.

What is a possible reason for this?

A. In the module di.xml, there were no default 3DS verification types configured as a VirtualType.
B. In the module config.xmi, the boolean value for can_capture was set to false.
C. In the module config.xmi, the node can_use_internal was not set to true.

Correct Answer: C
Section:
Explanation:

QUESTION 42

An Adobe Commerce developer is asked to implement a 15% surcharge for all users from a 'Wholesale' customer group. Keeping best practices in mind, what is a correct to accomplish this?

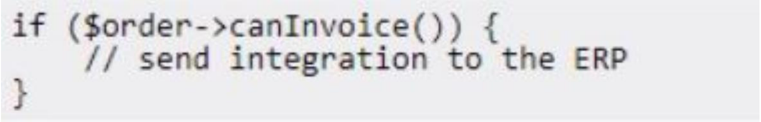
A. Declare a new total collector class to calculate the modified total if the current user is in the group, register it in the module's etc/sales .xml file, modify the checkout_cart_index.xml and checkout_index_index.xml layouts to include a new child in the totals block.

B. Create a Cart Price Rule that applies only to the 'Wholesale' group. Specify no conditions for the rule, and in the Actions section, specify for the rule to apply a 'Percent of product price discount', with the 'Discount Amount' field set to -15.

C. Create an Observer to the cataiog_product_get_final_price event. Check if the current customer is in the 'Wholesale' group, and if so, retrieve the product from the $observer->getEventC) data and Call $product->setData('final_price', $product->getData('final_price') * 1.15).

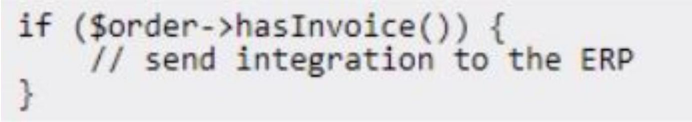
Correct Answer: C
Section:
Explanation:
The possible reason why the payment method is missing in the admin is that in the module config.xml, the node can_use_internal was not set to true. This node determines whether the payment method can be used in the admin area or not. If it is set to false or omitted, the payment method will not be available for admin orders. To enable the payment method for admin use only, the node can_use_internal should be set to true and the node can_use_checkout should be set to false. Verified
Reference: [Magento 2.4 DevDocs] [Magento Stack Exchange]

QUESTION 43
There is an integration developed using a cron service that runs twice a day, sending the Order ID to the integrated ERP system if there are orders that are able to create an invoice. The order is already loaded with the following code:
$order = $this->orderRepository->get($orderId);
In order to verify if the store has invoices to be created, what implementation would the Adobe Commerce developer use?
A)



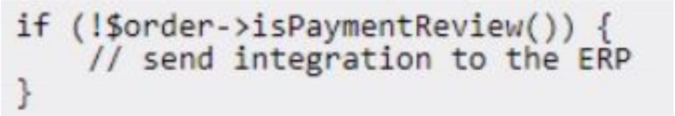
```
if ($order->canInvoice()) {
    // send integration to the ERP
}
```

B)



```
if ($order->hasInvoice()) {
    // send integration to the ERP
}
```

C)

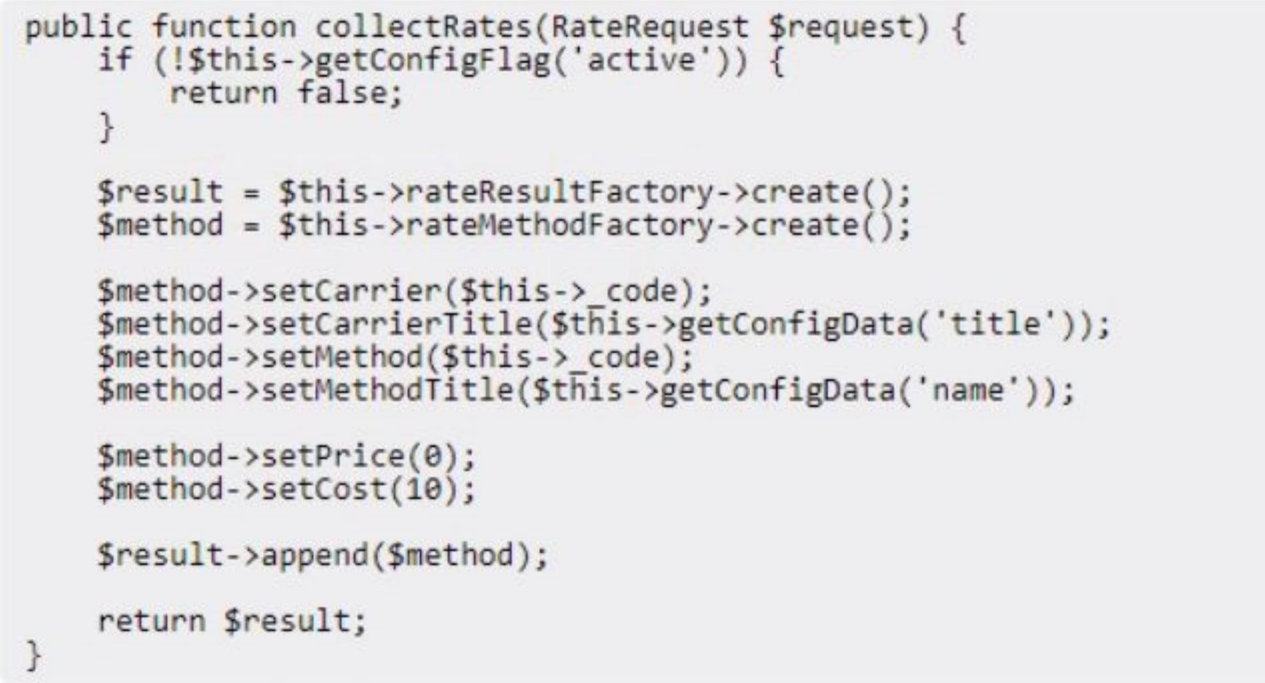


```
if (!$order->isPaymentReview()) {
    // send integration to the ERP
}
```

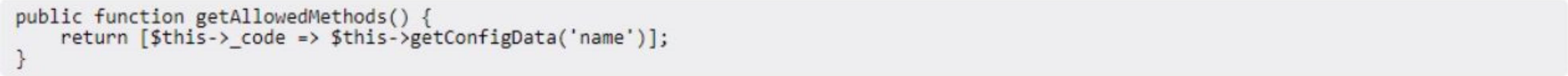
A. Option A
B. Option B
C. Option C

Correct Answer: B
Section:
Explanation:
The developer can use the canInvoice method of the order object to check if the order can be invoiced or not. This method returns true if the order has a state of new, processing, or payment review and has not been fully invoiced yet. The developer can use this method in a conditional statement to send the order ID to the ERP system only if the order can be invoiced. Verified
Reference: [Magento 2.4 DevDocs] [Magento Stack Exchange]

QUESTION 44
An Adobe Commerce developer has created a new shipping carrier Everything has been implemented and the collectRates() and getAllowedMethodsQ functions can be seen below:



```
public function collectRates(RateRequest $request) {
    if (!$this->getConfigFlag('active')) {
        return false;
    }

    $result = $this->rateResultFactory->create();
    $method = $this->rateMethodFactory->create();

    $method->setCarrier($this->_code);
    $method->setCarrierTitle($this->getConfigData('title'));
    $method->setMethod($this->_code);
    $method->setMethodTitle($this->getConfigData('name'));

    $method->setPrice(0);
    $method->setCost(10);

    $result->append($method);

    return $result;
}
```



```
public function getAllowedMethods() {
    return [$this->_code => $this->getConfigData('name')];
}
```

Given the above code, what would be the displayed cost of the shipping method and final amount charged to the customer?

A. The shipping method would display SO but customers would pay a $10 handling fee for their order.

B. The shipping method would display $0 and customers would pay $0 for using the new shipping method.

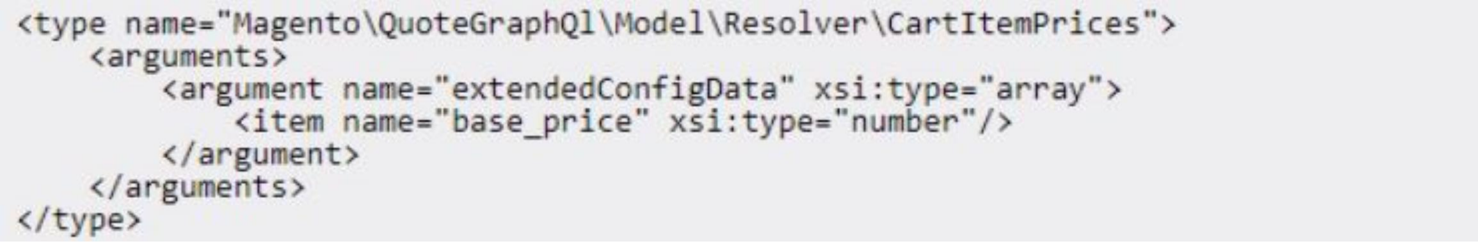
C. The shipping method would display $10 and customers would pay $10 for using the new shipping method.

Correct Answer: C
Section:
Explanation:
The shipping method would display $10 and customers would pay $10 for using the new shipping method. This is because the collectRates method sets the price and cost of the shipping method to 10 in the result object. The price is what is shown to the customer on the frontend and the cost is what is charged to the customer when they place the order. The handling fee is not used in this case because it is set to zero in the config.xml file. Verified
Reference: [Magento 2.4 DevDocs] [Magento Stack Exchange]

QUESTION 45
An Adobe Commerce developer is creating a new module to extend the functionality of the cart. The module is installed in app/code/CompanyName/ModuleName/.
How would an Adobe Commerce developer extend the existing CartItemPrices GraphQL schema to include a custom base_price field?

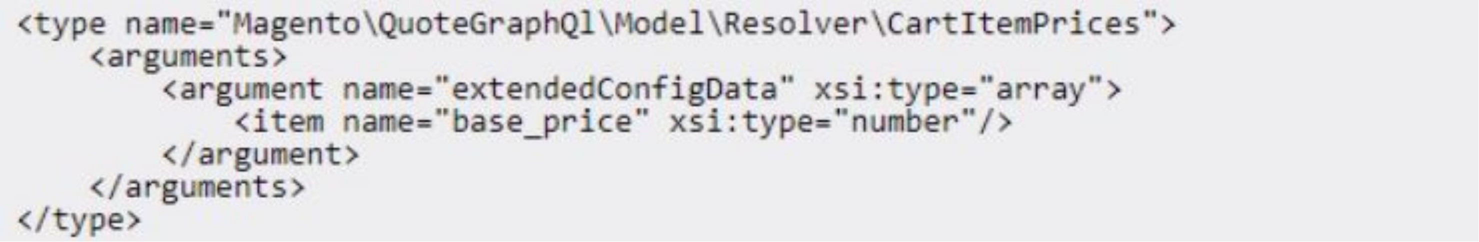
A. Create and Configure a for Hagento\QuoteGraphQl\Model\Resolver\CartItemPrices that adds the base_price field in the resolve() function.

B. Add the following to the module's etc/schema.graphqis file:



```
<type name="Magento\QuoteGraphQl\Model\Resolver\CartItemPrices">
    <arguments>
        <argument name="extendedConfigData" xsi:type="array">
            <item name="base_price" xsi:type="number"/>
        </argument>
    </arguments>
</type>
```

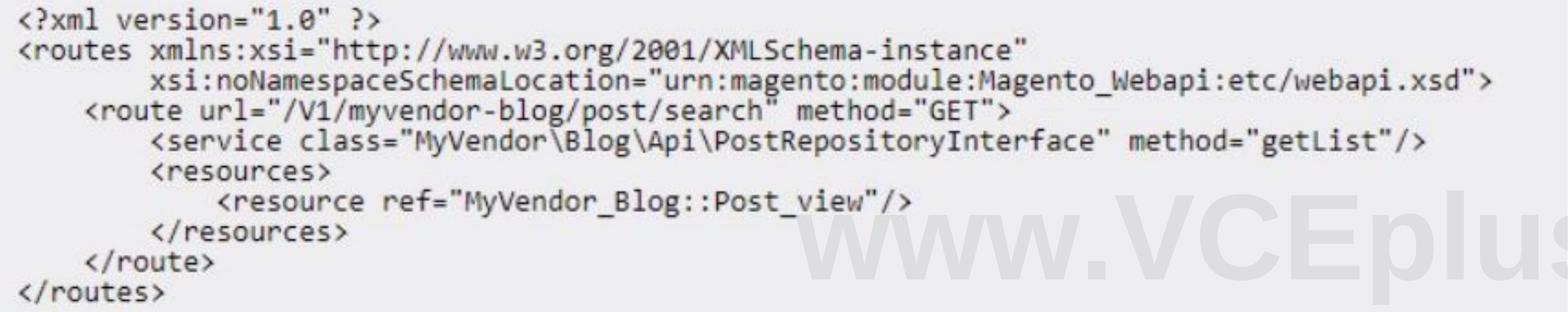
C. Add the following to the module's etc/graphqi/di.xmi file:



```
<type name="Magento\QuoteGraphQl\Model\Resolver\CartItemPrices">
    <arguments>
        <argument name="extendedConfigData" xsi:type="array">
            <item name="base_price" xsi:type="number"/>
        </argument>
    </arguments>
</type>
```

Correct Answer: B
Section:
Explanation:
The developer can extend the existing CartItemPrices GraphQL schema to include a custom base_price field by adding the following code to the module's etc/schema.graphqls file:
extend type CartItemPrices { base_price: Money! @doc(description: ''The base price of the cart item'') }
This code adds a new field called base_price to the CartItemPrices type and specifies that it is of type Money and it is not nullable. The @doc directive adds a description for the field that will be shown in the schema documentation. The developer also needs to create a custom resolver class for the base_price field and declare it in the di.xml file of the module. Verified
Reference: [Magento 2.4 DevDocs] [Magento Stack Exchange]

QUESTION 46
An Adobe Commerce developer added a new API method to search and retrieve a list of Posts for a custom Blog functionality. This is the content of the module's etc/webapi.xml file:



```
<?xml version="1.0" ?>
<routes xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:noNamespaceSchemaLocation="urn:magento:module:Magento_Webapi:etc/webapi.xsd">
    <route url="/V1/myvendor-blog/post/search" method="GET">
        <service class="MyVendor\Blog\Api\PostRepositoryInterface" method="getList"/>
        <resources>
            <resource ref="MyVendor_Blog::Post_view"/>
        </resources>
    </route>
</routes>
```

The new code has been deployed to production and the merchant is using https: //merchant. domain. com/swagger to review the new endpoint, but it is not visible in swagger.
What would be a reason for this?

A. The webapi.xml file should be moved into the etc/webapi_rest/webapi.xml file.
B. Since the new endpoint is not anonymous, the merchant needs to enter a valid integration token in swagger in order to see the new method.
C. The greturn annotation is missing in the MyVendor\Blog\Api\PostRepositoryInterf ace class.

Correct Answer: B
Section:
Explanation:
The reason why the new endpoint is not visible in swagger is that since the new endpoint is not anonymous, the merchant needs to enter a valid integration token in swagger in order to see the new method. The webapi.xml file specifies that the resource for the new endpoint is MyVendor_Blog::post, which means that only authorized users with this permission can access it. To generate an integration token, the merchant needs to create an integration in the admin panel and activate it. Then they can copy the token and paste it in swagger's authorization field. Verified
Reference: [Magento 2.4 DevDocs] [Magento Stack Exchange]

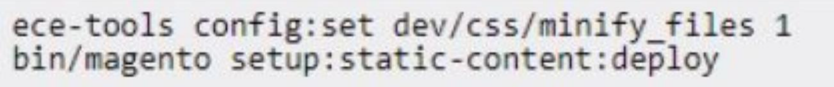
QUESTION 47
A message queue currently has queue/consumer-wait-for-messages set to true, which allows the consumer process to run until a message is inserted into the queue. A piece of functionality is driven by data stored in the model
\Magento\variable\Model\variable and this value is only loaded once during the consumer run. If the variable is updated we want the consumer to restart so that the new value is loaded into memory without having to reload the variable on each message consumed.
The Adobe Commerce developer has created an after plugin on the \Magento\Variable\Model\variable:: save() function.
How would the developer use the plugin to trigger the consumer restart?

A. Call the function \Magento\Framework\MessageQueue\PoisonPill\Poi5onPillPutInterface::put().

B. Call the function \Magento\Framework\MessageQueue\Consumers\TriggerRe5tartInterface:trigger().
C. Set the global Cache key trigger_consumer_restart t0 1.

Correct Answer: B
Section:
Explanation:
The developer can use the plugin to trigger the consumer restart by calling the function \Magento\Framework\MessageQueue\Consumers\TriggerRe5tartInterface:trigger(). This function will write a flag to the cache storage that will be checked by the consumer process.If the flag is set, the consumer process will terminate itself and restart with the updated configuration. Verified
Reference: [Magento 2.4 DevDocs]1

QUESTION 48
How would a developer enable the magnification of CSS files on an Adobe Commerce Cloud Staging environment?
A. Locally from the command line bin/magento config:set --lock-config dev/css/minify_files 1 Commit the app/etc/config.php file and redeploy.
B. Update the stores > setting > configuration > Advanced > Developer > css configuration in the Admin Panel.
C. SSH to the Adobe Commerce Staging environment. From the command line



```
ece-tools config:set dev/css/minify_files 1
bin/magento setup:static-content:deploy
```

Correct Answer: A
Section:
Explanation:
The developer can enable the magnification of CSS files on an Adobe Commerce Cloud Staging environment by locally running the command bin/magento config:set --lock-config dev/css/minify_files 1 from the command line. This will set the configuration value in the app/etc/config.php file and lock it from being changed in the Admin Panel.The developer then needs to commit the app/etc/config.php file and redeploy the environment. Verified
Reference: [Magento 2.4 DevDocs]2

QUESTION 49
A merchant of an Adobe Commerce Cloud project wants to setup one of their websites using a subdomain. The merchant is considering the domain to be set as secondstore.example.com.
In addition to editing the magento-vars.php file, and apply a domain check and set $_SERVER['MAGE_RUN_CODE'] and $_SERVER['MAGE_RUN_TYPE'].
What file is required to perform this action?
A. Configure secondstore.example.com subdomain route in NGINX virtual-host configuration file.
B. Configure secondstore.example.com subdomain route in .magento/services.yaml.
C. Configure secondstore.example.com subdomain route in .magento/routes.yaml.

Correct Answer: C
Section:
Explanation:
The developer can set up a subdomain for one of their websites by configuring the subdomain route in the .magento/routes.yaml file. This file defines how incoming requests are routed to different applications or services on the Adobe Commerce Cloud platform. The developer needs to add a route for secondstore.example.com and map it to the same application as example.com.The developer also needs to specify the upstream variable for secondstore.example.com as MAGE_RUN_CODE and MAGE_RUN_TYPE. Verified
Reference: [Magento 2.4 DevDocs]3

QUESTION 50
What is the command used to upgrade ECE-Tools on an Adobe Commerce Cloud platform?
A. php ./vendor/bin/ece-tools upgrade
B. composer update magento/ece-tools --with-all-dependencies
C. magento-cloud ece-tools:upgrade

Correct Answer: B
Section:

Explanation:

The command used to upgrade ECE-Tools on an Adobe Commerce Cloud platform is composer update magento/ece-tools --with-all-dependencies. This command will update the ECE-Tools package and its dependencies to the latest version available in the composer repository. The developer then needs to commit and push the changes to the composer.json and composer.lock files and redeploy the environment. Verified

Reference: [Magento 2.4 DevDocs]

QUESTION 51

An Adobe Commerce Cloud merchant has been experiencing significant downtime during production deployment. They have already checked that the application is in ideal state.

In addition to the configuration of the SCD.MATRIX variable to reduce amount of unnecessary theme files, what would be the next steps to reduce the downtime?

A. 1. Check SCD is configured under the build phase. 2. Increase the SCD.THREADS to speed up the build process.

B. 1. Check SCD is configured under deploy phase. 2. Decrease the SCD.THREADS to speed up the build process

C. 1. Check SCD is configured under the build phase. 2. Check if Adobe Commerce Cloud automatically adjusts SCD.THREADS.

Correct Answer: A

Section:

Explanation:

The next steps to reduce the downtime are to check that the SCD is configured under the build phase and to increase the SCD.THREADS to speed up the build process. The SCD stands for static content deployment, which is the process of generating and deploying static files such as CSS, JS, images, etc. By configuring the SCD under the build phase, the static files are generated before the code is deployed to the production environment, which reduces the downtime during deployment. The SCD.THREADS is a variable that determines how many threads are used for parallel processing during the SCD.By increasing the SCD.THREADS, the developer can improve the performance and efficiency of the SCD process. Verified

Reference: [Magento 2.4 DevDocs]12

QUESTION 52

What are two ways to access the PHP error logs on Adobe Commerce Cloud? (Choose Two.)

A. Use the dedicated command from Cloud CLI for Commerce.

B. Navigate to the dedicated entry in the Project Web Interface.

C. Connect to the the servers via SSH and localize the log files.

D. Use the Adobe Admin Log application.

Correct Answer: A, C

Section:

Explanation:

Two ways to access the PHP error logs on Adobe Commerce Cloud are to use the dedicated command from Cloud CLI for Commerce and to connect to the servers via SSH and localize the log files. The Cloud CLI for Commerce is a command-line tool that allows developers to interact with their Adobe Commerce Cloud projects and environments. The developer can use the command magento-cloud log php to view or download the PHP error logs from any environment.Alternatively, the developer can connect to the servers via SSH and navigate to the var/log directory where the PHP error logs are stored. Verified

Reference: [Magento 2.4 DevDocs]3

QUESTION 53

On an Adobe Commerce Cloud platform, in which order does the ECE-Tools package apply patches?

A. 1. All required Magento patches included in the Cloud Patches for Commerce package. 2. Custom patches in the /m2-hotfixes directory in alphabetical order by patch name. 3. Selected optional Magento patches included in the Quality Patches Tool.

B. 1. All required Magento patches included in the Cloud Patches for Commerce package. 2. Selected optional Magento patches included in the Quality Patches Tool. 3. Custom patches in the /m2-hotfixes directory in alphabetical order by patch name.

C. 1. Custom patches in the /m2-hotfixes directory in alphabetical order by patch name. 2. All required Magento patches included in the Cloud Patches for Commerce package. 3. Selected optional Magento patches included in the Quality Patches Tool.

Correct Answer: B

Section:

Explanation:

The order in which the ECE-Tools package applies patches is as follows:

All required Magento patches included in the Cloud Patches for Commerce package.
Selected optional Magento patches included in the Quality Patches Tool.
Custom patches in the /m2-hotfixes directory in alphabetical order by patch name.
The ECE-Tools package is a set of scripts and tools designed to manage and deploy Adobe Commerce Cloud projects. The Cloud Patches for Commerce package is a dependency of ECE-Tools that provides a set of required patches for Magento core issues that affect Adobe Commerce Cloud functionality. The Quality Patches Tool is an optional tool that allows developers to apply individual patches for specific Magento issues without waiting for a full product release. The /m2-hotfixes directory is a directory where developers can place their own custom patches for their Adobe Commerce Cloud projects. Verified
Reference: [Magento 2.4 DevDocs]

QUESTION 54
When attempting operations that require lengthy processing, a merchant on Adobe Commerce Cloud receives a timeout error after 180 seconds.
How would the developer deal with this issue?
A. 1. Modify admin timeout into .magento.app.yamifile. 2. Commit and push that code from the local environment. 3. Move code to Production environment.
B. 1. In the Fastly Configuration section > Advanced Configuration. 2. Set the Admin path timeout value in seconds. 3. Save config and Upload VCL to Fastly.
C. 1. Modify admin timeout into app/etc/config.php file. 2. Commit and push that code from the local environment. 3. Submit a support ticket to apply the changes.

Correct Answer: B
Section:
Explanation:
The developer can deal with this issue by modifying the admin path timeout value in seconds in the Fastly Configuration section > Advanced Configuration in the Admin Panel. Fastly is a cloud-based caching service that improves site performance and security for Adobe Commerce Cloud projects. Fastly has a default timeout value of 180 seconds for admin requests, which means that any request that takes longer than 180 seconds will be terminated and result in a timeout error. The developer can increase this value to allow longer processing time for admin requests without causing errors. The developer also needs to save the configuration and upload VCL to Fastly to apply the changes. Verified
Reference: [Magento 2.4 DevDocs]

QUESTION 55
An Adobe Commerce Cloud project is using Enhanced Integration Environments with two install a new payment module.
The developer is using Cloud CLI for Commerce tool.
What would a developer do to test this new feature under the integration environment?
A. 1. Duplicate one of the integration environment branches. 2. Create a new active branch from integration and install the module. 3. Push the changes.
B. 1. Create a new branch from integration and install the module. 2. Push the changes. 3. Branch active status check is not necessary.
C. 1. Deactivate one of the active integration environment branches. 2. Create a new active branch from integration and install the module. 3. Push the changes.

Correct Answer: C
Section:
Explanation:
The developer can test the new feature under the integration environment by deactivating one of the active integration environment branches, creating a new active branch from integration and installing the module, and pushing the changes. This is because Enhanced Integration Environments have a limit of four active branches at a time, and each branch has its own dedicated database and services.The developer can use the Cloud CLI for Commerce tool to manage the branches and deploy the code changes. Verified
Reference: [Magento 2.4 DevDocs]1

QUESTION 56
ECE-Tools provides a set of tools that can be used to manage and maintain your Adobe Commerce Cloud environment. What are some of the features provided by ECE-Tools?
A. Builds application, Applies custom patches and Dump configuration for static content deployment.
B. Fastly configuration, Applies custom patches and Dump configuration for static content deployment.
C. Builds application, Applies custom patches, and Shows the list of S3 backup tar.gz files.

Correct Answer: A
Section:
Explanation:
Some of the features provided by ECE-Tools are building application, applying custom patches, and dumping configuration for static content deployment. ECE-Tools is a set of scripts and tools designed to manage and

deploy Adobe Commerce Cloud projects.It provides commands for building application code, applying patches for Magento core issues or custom modules, and dumping configuration settings for static content deployment optimization. Verified
Reference: [Magento 2.4 DevDocs]2

QUESTION 57

A developer is working on an Adobe Commerce Cloud project and wants to get connection data for the environment's deployed services. The developer has all of the necessary permissions to do this.
Which two options would the developer take to get the connection credentials? (Choose Two.)

A. Run the magento-cloud relationships CLI Command.
B. Get the data from the Project Web Interface dedicated section.
C. Execute ece-tools env:config:show services Command.
D. Connect to server via SSH and read $_ENV['services'] variable.

Correct Answer: A, D
Section:
Explanation:

Two options to get the connection credentials for the environment's deployed services are to run the magento-cloud relationships CLI command and to connect to the server via SSH and read $_ENV['services'] variable. The magento-cloud relationships CLI command displays information about the relationships between an environment and its services, such as database, cache, search, etc. The developer can use this command to get the connection data for each service in JSON format.Alternatively, the developer can connect to the server via SSH and read the $_ENV['services'] variable, which contains the same information as the CLI command output. Verified
Reference: [Magento 2.4 DevDocs]3

QUESTION 58

A merchant is experiencing performance issues on integration environments of their Adobe Commerce Cloud Pro plan and wants to upgrade to Enhanced Integration Environments.
What are the steps necessary prior to redeploying in order to upgrade to Enhanced Integration Environments?

A. 1. Limit the number of Integration branches to two 2. Submit a support ticket requesting the upgrade
B. 1. Limit the number of Integration branches to three 2. Set the ENV.ENVIRONMENT in .magento.env.yaml to ENHANCEDJNTEGRATION
C. 1. Limit the number of Integration branches to four 2. Configure integration environments in the cloud GUI and set the Enhanced switch to On

Correct Answer: A
Section:
Explanation:

The steps necessary prior to redeploying in order to upgrade to Enhanced Integration Environments are to limit the number of integration branches to two and to submit a support ticket requesting the upgrade. Enhanced Integration Environments are an improved version of integration environments that offer better performance, stability, and security. They have a limit of four active branches at a time, but only two branches can be migrated from standard integration environments. The developer needs to delete or deactivate any extra branches before requesting the upgrade from Adobe support. Verified
Reference: [Magento 2.4 DevDocs]

QUESTION 59

How would a developer turn on outgoing emails on an Adobe Commerce Cloud Staging environment?

A. From the command line ece-tools enable_smtp true
B. From the command line magento-cloud environment:info -p -e <environment-id> enable_smtp true
C. Access the Project Web Interface and select the Staging environment. Select Configure environment. Toggle Outgoing emails On

Correct Answer: C
Section:
Explanation:

The developer can turn on outgoing emails on an Adobe Commerce Cloud Staging environment by accessing the Project Web Interface and selecting the Staging environment. Then, the developer can select Configure environment and toggle Outgoing emails On.This will enable the SMTP service for the Staging environment and allow emails to be sent from the application. Verified
Reference: [Magento 2.4 DevDocs]1

QUESTION 60

On an Adobe Commerce Cloud platform, at what level is the variable env: composer_auth located in the Project Web Interface?
A. In the Environment-specific variables.
B. In the Integration variables.
C. In the Project variables.

Correct Answer: C
Section:
Explanation:
The variable env: composer_auth is located in the Project variables section in the Project Web Interface. This variable is used to store the authentication credentials for Composer repositories that require access keys or tokens.The developer can set this variable at the project level to apply it to all environments, or override it at the environment level if needed. Verified
Reference: [Magento 2.4 DevDocs]2

QUESTION 61
What are two features with Adobe Commerce Cloud that come out of the box? (Choose Two.)
A. Support ACL
B. Continuous deployment provided with the platform
C. A built in connector with all major blog platforms
D. Fastly

Correct Answer: A, D
Section:
Explanation:
Two features that come out of the box with Adobe Commerce Cloud are Support ACL and Fastly. Support ACL is a feature that allows the developer to manage access control lists for different users and roles on the Adobe Commerce Cloud platform. The developer can create and assign permissions for different actions and resources on the project and environment levels. Fastly is a cloud-based caching service that improves site performance and security for Adobe Commerce Cloud projects.Fastly provides features such as CDN, image optimization, WAF, DDoS protection, etc. Verified
Reference: [Magento 2.4 DevDocs]3

QUESTION 62
A new customer registered on the Integration environment of an Adobe Commerce Cloud project but did not receive a welcome email What would be blocking the email from being sent?
A. SendGrid has not been configured for this environment.
B. On all Integration environments, email is always disabled.
C. The Outgoing Emails setting is disabled into Environment Settings in the Project Web Interface.

Correct Answer: C
Section:
Explanation:
The reason why the new customer did not receive a welcome email is that the Outgoing Emails setting is disabled in the Environment Settings in the Project Web Interface. This setting controls whether emails are sent from the application or not. By default, this setting is disabled for integration environments to prevent spamming or testing emails from being sent to real customers or recipients. The developer can enable this setting if they want to test email functionality on integration environments. Verified
Reference: [Magento 2.4 DevDocs]

QUESTION 63
A developer wants to deploy a new release to the Adobe Commerce Cloud Staging environment, but first they need the latest code from Production.
What would the developer do to update the Staging environment?
A. 1. Log in to the Project Web Interface. 2. Choose the Staging environment, and click Sync
B. 1. Log in to the Project Web Interface. 2. Choose the Staging environment, and click Merge
C. 1. Checkout to Production environment 2. Use the magento-cloud synchronize <environment-ID> Commerce CLI Command

Correct Answer: A
Section:
Explanation:

The developer can update the Staging environment with the latest code from Production by logging in to the Project Web Interface, choosing the Staging environment, and clicking Sync. This will synchronize the code, data, and media files from Production to Staging, creating an exact copy of Production on Staging. The developer can then deploy the new release to Staging and test it before pushing it to Production. Verified
Reference: [Magento 2.4 DevDocs]

QUESTION 64
A Project Architect needs to add a new developer who needs to be able to push code in an Adobe Commerce Cloud project. No integration with a third-party repository provider is setup.
What two actions would be required to ensure the developer has access? (Choose Two.)
A. The developer's SSH public key must be added into a file named ~/.ssh/authorized_keys
B. The developer needs to add SSH public key in the Cloud Account dashboard settings
C. The developer's email must be added under Users in the Cloud Project Web UI
D. The Adobe Commerce admin user must be created and the developer's SSH public key must be added on their local system

Correct Answer: B, C
Section:
Explanation:
To ensure the developer has access to push code in an Adobe Commerce Cloud project, the developer's email must be added under Users in the Cloud Project Web UI and the developer needs to add SSH public key in the Cloud Account dashboard settings. The Cloud Project Web UI is a web interface that allows managing and configuring Adobe Commerce Cloud projects and environments. The developer's email must be added under Users to grant them access to the project and assign them a role and permissions. The Cloud Account dashboard settings is a web interface that allows managing and configuring Adobe Commerce Cloud accounts and SSH keys. The developer needs to add SSH public key in the settings to enable secure connection to the project and environments via SSH. Verified
Reference: [Magento 2.4 DevDocs]

QUESTION 65
An Adobe Commerce Cloud developer wants to be sure that, even after transferring database from Production to Staging, the payment configurations are still valid on the Staging environment.
What does the developer need to add to be sure that the configurations are always properly set?
A. Lines in the dedicated core_conf ig_data_stg table.
B. Project level environment variables.
C. Environment level environment variables.

Correct Answer: C
Section:
Explanation:
The developer needs to add environment level environment variables to be sure that the payment configurations are always properly set on the Staging environment. Environment variables are configuration settings that affect the behavior of the Adobe Commerce Cloud application and services. Environment variables can be set at the project level or the environment level. Project level variables apply to all environments, while environment level variables override the project level variables for a specific environment. The developer can use environment level variables to customize the payment configurations for the Staging environment without affecting other environments. Verified
Reference: [Magento 2.4 DevDocs]

QUESTION 66
During database migration in the Adobe Commerce Cloud integration environment, a developer experienced a disk space error causing the database import to fail.
How would the developer fix this issue?
A. Increase the disk space of the database service.
B. Add a new database node and enable split database.
C. Change the database engine to PostgreSQL that has no disk space limit.

Correct Answer: A
Section:
Explanation:
The developer can fix this issue by increasing the disk space of the database service. The database service is one of the services that run on the Adobe Commerce Cloud platform and provide functionality for the application. The database service uses MySQL as the database engine and stores data for products, customers, orders, etc. The disk space of the database service determines how much data can be stored and processed by the database. If the disk space is insufficient, the database import can fail with a disk space error. The developer can increase the disk space of the database service by modifying the

.magento/services.yaml file and redeploying the environment. Verified
Reference: [Magento 2.4 DevDocs]

QUESTION 67
On an Adobe Commerce Cloud platform, what type of environment will be provisioned when launching the CLI for Commerce command magento-cloud environment:branch <environment-name> ?
A. An empty integration environment without any code or database.
B. An integration environment with fresh Adobe Commerce Cloud installation.
C. An integration environment with the code and database from the parent environment.

Correct Answer: C
Section:
Explanation:
The type of environment that will be provisioned when launching the CLI for Commerce command magento-cloud environment:branch <environment-name> is an integration environment with the code and database from the parent environment. Integration environments are temporary environments that are used for testing and development purposes on the Adobe Commerce Cloud platform. They can be created from any branch of code and have their own dedicated database and services. When creating an integration environment using the CLI for Commerce command, the code and database from the parent environment are copied to the new integration environment, creating an exact replica of the parent environment. Verified
Reference: [Magento 2.4 DevDocs]

QUESTION 68
For security reasons, merchant requested to a developer to change default admin url to a unique url for every branch/environment of their Adobe Commerce Cloud project.
Which CLI command would the developer use update the admin url?
A. ece-tools variable:update ADMIN_URL
B. magento-cloud variable:set ADMIN_URL
C. bin/magento adminuri:set <admin_uri>

Correct Answer: B
Section:
Explanation:
The CLI command that the developer would use to update the admin url is magento-cloud variable:set ADMIN_URL. This command sets an environment variable called ADMIN_URL with a custom value for the admin url on a specific environment. Environment variables are configuration settings that affect the behavior of the Adobe Commerce Cloud application and services. By setting an environment variable for ADMIN_URL, the developer can change the default admin url to a unique url for every branch/environment of their Adobe Commerce Cloud project. Verified
Reference: [Magento 2.4 DevDocs]

QUESTION 69
An Adobe Commerce developer is about to deploy a critical feature to their Adobe Commerce Cloud (Pro Plan) production. They want to create a snapshot in order to be able to rollback if there is an issue with the feature.
How would they create the snapshot?
A. Use the dedicated button on Project Web Interface.
B. Use the Cloud CLI for Commerce dedicated command.
C. Create a ticket to Adobe Commerce Cloud support.

Correct Answer: A
Section:
Explanation:
The developer can create a snapshot before deploying a critical feature to their Adobe Commerce Cloud (Pro Plan) production by using the dedicated button on Project Web Interface. A snapshot is a backup of an entire environment, including code, data, media files, and configuration settings. A snapshot can be used to restore an environment to a previous state in case of any issues or errors during deployment or testing. The developer can create a snapshot by accessing the Project Web Interface, choosing an environment, and clicking Create Snapshot. Verified
Reference: [Magento 2.4 DevDocs]