Adobe

Exam Questions AD0-E724

Adobe Commerce Developer Professional

NEW QUESTION 1
Which Adobe Commerce table stores all cron data?

A. schedule
B. cronjob
C. cron_schedule

Answer: C

Explanation:
The Adobe Commerce table that stores all cron job data iscron_schedule. This table maintains records of all scheduled cron tasks, including their statuses, execution times, and any messages related to their execution. It plays a central role in Magento's scheduling system, allowing for themanagement and monitoring of background tasks that are essential for various system functions and integrations.

NEW QUESTION 2
Which file is used to add a custom router class to the list of routers?

A. routes.xml
B. di.xml
C. config.xml

Answer: A

Explanation:
To add a custom router class to the list of routers in Magento, theroutes.xml file is used. This file should be located in theetcdirectory of the module, under the appropriate area (eitherfrontendoradminhtml). Within theroutes.xmlfile, you define a router with an ID, a route with an ID and frontName, and specify the module that the route corresponds to. This setup allows Magento to recognize and utilize the custom router when processing URLs, directing requests to the appropriate controllers based on the custom routing logic defined.

NEW QUESTION 3
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

Answer: C

NEW QUESTION 4
How can a developer prioritize a plugin's execution, if possible?

A. The developer can use sortOrder property by specifying a lower value than the target plugin.
B. The developer can use sortOrder property by specifying a higher value than the target plugin.
C. This cannot be achieved as the plugins are always executed by their module's load order in app/etc/config.php file.

Answer: A

Explanation:
A developer can prioritize the execution of a plugin by using thesortOrder property within the plugin's declaration in thedi.xmlfile. Specifying a lower value for the sortOrderproperty gives the plugin higher priority, meaning it will be executed before other plugins with a higher sortOrder value. This allows developers to control the order of plugin execution, which can be critical for ensuring the desired outcome when multiple plugins are affecting the same method.

NEW QUESTION 5
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 inde
B. It is needed since attribute identity is set to true.
C. Column (entity_id) does not have inde
D. It is needed since attribute identity is set to false.
E. Column (student_name) does not have attribute length.
F. null is not a valid value for column (roll_no).

Answer: AC

Explanation:
The correct answers are A and C. The errors in the snippet are:
? Column roll_no does not have an index. It is needed since attribute_identity is set to true.
? Column student_name does not have an attribute length. Theattribute_identityattribute specifies whether the primary key of the table should be auto-incremented. Ifattribute_identityis set totrue, then theroll_nocolumn must have an index. Thestudent_namecolumn 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>
<primary_key>true</primary_key>
<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.

NEW QUESTION 6
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

Answer: A

Explanation:
The correct implementation to check if an order is eligible for invoicing is to use the $order->canInvoice() method. This method checks whether the order meets all necessary conditions for an invoice to be created, such as the order not being fully invoiced or canceled.
Option A is correct for the following reasons:
? Using canInvoice() for Invoicing Eligibility:The $order->canInvoice() method is specifically designed to verify if an order can have an invoice generated. It returns true only if the order is in a state where it can be invoiced. This makes it the appropriate method for determining whether the order should be sent to the ERP system for invoicing.
? uk.co.certification.simulator.questionpool.PList@45e8e59a
: Magento??s developer documentation on the Order model highlights canInvoice() as the recommended approach for determining invoice eligibility, particularly when automating processes like ERP integration.
Alternatives and Limitations:
Option B: The $order->hasInvoice() method only checks if there is already an invoice associated with the order, which does not indicate whether the order is eligible for new invoicing. It returns true if any invoice exists for the order, which is not suitable for this scenario.
Option C: The $order->isPaymentReview() method checks if the order is in a payment review state, which is not directly related to invoice creation eligibility. It would not provide accurate information on whether the order can be invoiced.
By using canInvoice(), the developer ensures that the cron job will only send orders that are ready for invoicing to the ERP system, adhering to Adobe Commerce??s order processing logic.

NEW QUESTION 7
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.

Answer: C

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 References: https://devdocs.magento.com/guides/v2.3/extension-dev-guide/plugins.html

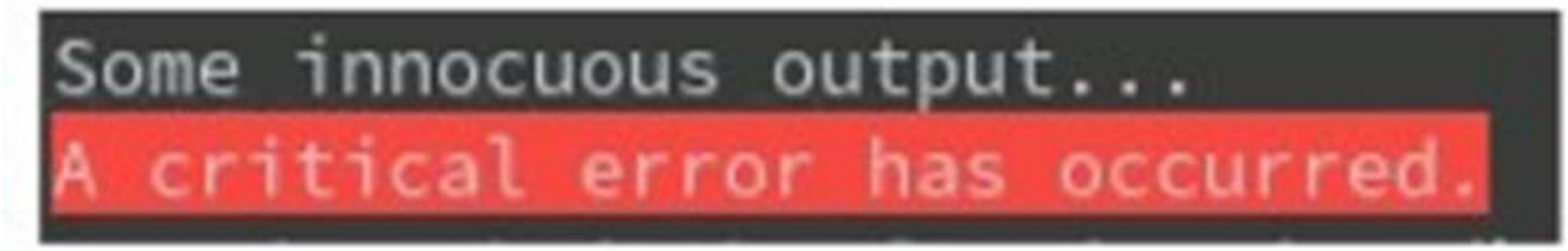
NEW QUESTION 8
A developer wants to deploy a new release to the Adobe Commerce Cloud Staging environment, but first they need the latest code from Production.
What would the developer do to update the Staging environment?

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

Answer: A

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 References: [Magento 2.4 DevDocs]

NEW QUESTION 9
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):



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.

Answer: B

Explanation:
In Adobe Commerce, when developing custom console commands, you can customize output messages by using special tags provided by Symfony Console, which Adobe Commerce relies on. These tags are designed to help differentiate types of messages and can be used to add color or emphasis to the output, enhancing visibility. For critical error messages, wrapping the message in the <error> tag will display it in red, as shown in your screenshot. The available tags include:
? <error> for red-colored error messages.
? <info> for informational messages (often displayed in blue).
? <comment> for comments or warnings (usually yellow).
$output->writeln('<error>A critical error has occurred.</error>');
This method is effective and widely used for output customization in Symfony-based console commands.
Additional Resources:
? Adobe Commerce Developer Guide: Console Command Customization
? Symfony Console Output Formatting

NEW QUESTION 10
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

Answer: BD

Explanation:
Adobe Commerce Cloud offers several out-of-the-box features, including built-in Fastly integration for CDN and web application firewall services, as well as continuous deployment capabilities through its cloud infrastructure.
? Continuous Deployment:
? uk.co.certification.simulator.questionpool.PList@200a841f
? Fastly Integration:
? Why Options B and D are Correct:
:
Adobe Commerce Cloud documentation onFastly CDN
Adobe Commerce Cloud documentation onDeployment and CI/CD

NEW QUESTION 10
Under which section should the soft dependency for a module be listed in app/code/<Vendor>/<Module>/composer.json file?

A. suggest*: {
B. }optional": {
C. }soft": {
D. }

Answer: A

Explanation:
Soft dependencies for a module should be listed under the "suggest" section in thecomposer.jsonfile of the module. This section is used to list packages that enhance or work well with the module but are not strictly required for the module to function. By using the "suggest" section, developers can inform others about optional packages that could improve functionality or integration with the module, without making them mandatory dependencies.

NEW QUESTION 12
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.

Answer: A

Explanation:
To minimize downtime during deployment, one of the most effective strategies is to configure static content deployment (SCD) to run during the build phase and optimize the number of threads used during the process.
? Configuring SCD in the Build Phase:
? uk.co.certification.simulator.questionpool.PList@22cc61b1
? Increasing SCD.THREADS:
? Why Option A is Correct:
:Adobe Commerce Cloud documentation onSCD Configuration

NEW QUESTION 13
What is one way a developer can upgrade the ECE-Tools package on an Adobe Commerce Cloud project?

A. Cloud CLI for Commerce tool
B. Project Web Interface
C. Composer

Answer: C

Explanation:
According to the Adobe Commerce Developer Documentation, one way a developer can upgrade the ECE-Tools package on an Adobe Commerce Cloud project is by using Composer, which is a dependency management tool for PHP projects. The ECE- Tools package contains scripts and tools that help manage and deploy Adobe Commerce Cloud projects on the cloud infrastructure. To upgrade the ECE-Tools package using Composer, the developer needs to run the following command in the project root directory: composer update magento/ece-tools --with-dependencies
On an Adobe Commerce Cloud project, the recommended way to upgrade the ECE-Tools package is through Composer, a dependency manager for PHP. Composer is used to manage the dependencies of the project, including ECE-Tools, and it provides the necessary commands to update packages to their latest versions or to specific versions as required.

NEW 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

Answer: BC

Explanation:
The file type within system.xml instructs Magento to render a file input field. This is crucial for allowing users to upload a file in the admin configuration, which then can be processed by the backend model.
References: The Magento official documentation outlines how system.xml configuration files can be used to add custom fields to the system configuration in the Magento backend.
Options A and D are incorrect because they do not specifically address the requirements of setting up a file upload field and processing a CSV import. Option A configures a backend model but lacks the necessary file type definition. Option D defines a custom type for the file field but does not directly address the file upload or CSV processing requirements.

NEW QUESTION 22
A custom theme Is being developed in the Adobe Commerce store, and the developer needs to override the current parent theme styles.
Which less file should the developer use to achieve this goal?

A. _theme.override.less
B. _hemeless
C. _source.less

Answer: B

Explanation:
To override the current parent theme styles in a custom theme being developed for Adobe Commerce, the developer should use the_theme.lessfile. This file is specifically designed for customizing and overriding the default styles provided by the parent theme, making option B the correct choice. The_theme.lessfile is a central place for theme-specific customizations.

NEW QUESTION 25
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: Vendor\FreeDownload\Model\Download Vendor\FreeDownload\Model\DownloadPdf extends Vendor\FreeDownload\Model\Download
Vendor\FreeDownload\Model\DownloadImage extends Vendor\FreeDownload\Model\Download
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

Answer: C

Explanation:
To configure a parameter for a parent class so that it propagates to all descendant classes, the correct approach is to define the parameter on the parent class within di.xml. This way, all child classes inheriting from this parent will automatically use the parameter value unless explicitly overridden.
Option C is correct for the following reasons:
? Configuring on the Parent Class (Vendor\FreeDownload\Model\Download):By setting the tracking_level parameter directly on the Download class, you ensure that all classes extending this class, such as DownloadPdf andDownloadImage, will inherit the tracking_level parameter value. This method leverages Magento's dependency injection configuration, which allows parameters set on a parent class to cascade to child classes.
? uk.co.certification.simulator.questionpool.PList@15a6494
: Magento??s official developer documentation outlines that class dependencies and configuration parameters defined in di.xml at a higher level are accessible to descendant classes. This is a standard practice in Magento for setting parameters that affect a hierarchy of classes.
Avoiding Redundant Configuration:Unlike Option A, which sets the parameter on an individual child class, or Option B, which redundantly sets the parameter on multiple child classes, Option C is optimal as it centralizes the configuration. This reduces the risk of discrepancies and simplifies maintenance.
Options A and B are incorrect because:
Option A configures the parameter on a single child class, which will not affect other child classes such as DownloadImage.
Option B redundantly sets the parameter for each child class individually, which is unnecessary when the parameter can be inherited from the parent.

NEW QUESTION 27
A developer is making customizations in the checkout, and access to the quotes shipping address is needed. Which file provides the shipping address of the current quote?

A. Magento_Checkout/js/model/quote
B. Magento_Quote/js/model/model
C. Magento_Checkout/js/model/quote-shipping-address

Answer: A

Explanation:
This file provides the shipping address of the current quote by using the getShippingAddress() method. For example, the following code snippet gets the shipping address from the quote object and logs it to the console:
define(['Magento_Checkout/js/model/quote'
],function(quote) { 'use strict';
varshippingAddress = quote.getShippingAddress(); console.log(shippingAddress);
});
The file Magento_Quote/js/model/model does not exist in Magento 2, and the file Magento_Checkout/js/model/quote-shipping-address is not a valid way to access the shipping address of the current quote. You can read more about the quote object and its methods in the Magento 2 developer documentation.
In Adobe Commerce, the shipping address of the current quote is accessed through the JavaScript fileMagento_Checkout/js/model/quote. This file includes various quote-related data, including shipping and billing addresses, items in the cart, and totals. There is no Magento_Checkout/js/model/quote-shipping-addressfile, and Magento_Quote/js/model/modelis not a valid path, making option A the correct choice.

NEW QUESTION 30
Which is a correct CMS content element in Adobe Commerce?

A. Widget
B. Sheet
C. Image

Answer: A

Explanation:
A widget is a CMS content element that can be used to display dynamic content on a page. Widgets can be used to display things like product reviews, social media feeds, or even custom content.
In Adobe Commerce, widgets are a correct CMS content element. Widgets allow merchants to add dynamic data or content blocks to CMS pages, static blocks, and various other locations throughout the store's layout without needing to directly edit the site's code. Options B (Sheet) and C (Image) are not recognized CMS content elements in the context of Adobe Commerce's terminology, making option A the correct answer.

NEW QUESTION 35
What database engine is part of the infrastructure of Adobe Commerce Cloud projects?

A. Percona
B. MariaDB
C. MySQL

Answer: B

Explanation:
The database engine that is part of the infrastructure of Adobe Commerce Cloud projects is MariaDB. MariaDB is a fork of MySQL that offers improved performance, scalability, and security features.
The database engine that is part of the infrastructure of Adobe Commerce Cloud projects is MariaDB. Adobe Commerce Cloud is configured to use MariaDB, which is a binary drop-in replacement for MySQL and is chosen for its performance, reliability, and feature set.

NEW QUESTION 38
A merchant has noticed an error in the checkout. The accessed URL is /checkout.
Where can the developer find the responsible controller in the Magento.Checkout module?

A. Controller/Index/Index.php
B. Controller/Index/Checkout.php
C. Controller/Checkout/Index.php

Answer: C

Explanation:
The controller responsible for handling the /checkout URL is located in Controller/Checkout/Index.php in the Magento.Checkout module1. This controller extends from \Magento\Checkout\Controller\Index\Index, which implements the execute() method that renders the checkout page1.
In the Magento_Checkout module, the responsible controller for the /checkout URL can be found inController/Checkout/Index.php. This controller handles the index action for the checkout process, initiating the checkout page when a customer navigates to /checkout. The organization of controllers in Magento follows a convention where the URL path corresponds to the directory structure within the module, making it easier to locate and understand the functionality associated with specific routes.

NEW QUESTION 40
Which CLI command should be used to determine that static content signing is enabled?

A. bin/magento config:show dev/static/status
B. bin/magento config:show dev/static/sign
C. bin/magento config:show dev/static/sign/status

Answer: B

Explanation:
After a thorough search of the provided documents, I couldn't find a direct reference to the specific CLI command for determining if static content signing is enabled in Magento. However, the typical command for checking configuration settings in Magento is bin/magento config:show <path>, where<path>is the configuration path for the setting you wish to view. Based on Magento's configuration path patterns and the options provided, the most logical choice would beB. bin/magento config:show dev/static/sign, although this cannot be confirmed without further context or documentation.

NEW QUESTION 41
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

Answer: C

Explanation:
To prevent the media_gallery attribute from being exported as part of the custom entity's data, the developer should add this attribute to the $_disabledAttrs class property array. This effectively excludes the attribute from the export process, ensuring that it does not appear in the exported file and thus will not be modified or re-imported inadvertently.
Option C is correct for the following reasons:
? Preventing Export with $_disabledAttrs:Adding media_gallery to the
$_disabledAttrs array tells the system to skip this attribute during export. This prevents the attribute from being included in the export file, thus removing the risk of it being altered by external software that handles the file. Since the attribute will not be present in the exported data, it will remain unchanged in the database when re-importing.
? uk.co.certification.simulator.questionpool.PList@6993fe6
: The Adobe Commerce documentation on import/export processes outlines how
$_disabledAttrs can be used to filter out sensitive or unnecessary attributes from export files.

Alternative Options and Their Limitations:
Option A: Using $_transformAttrs with a serializer is useful for encoding or decoding attribute data during export/import, but it does not prevent the attribute from being included in the export. This would only help if the media_gallery data needed transformation, not exclusion.
Option B: $_strippedAttrs is applicable for filtering attributes from the imported file, not the exported file. It would not stop the attribute from being included in the export and hence does not align with the need to prevent modifications during export.
By configuring $_disabledAttrs, the developer effectively ensures the media_gallery attribute remains unmodified by preventing it from being included in export files altogether.

NEW QUESTION 46
An Adobe Commerce developer wants to generate a list of products using ProductRepositorylnterf 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 passe
B. Write an event observer to 0 listen for the event cataiog_product_coiiection_ioad_befor
C. Iterate through the registered search criteria, and if found, apply the needed join and filter to the events scollection.
D. Add a CUStOm filter to the Virtual type "agento\Catalog\Model\Api\SearchCriteria\CollectionProcessor\ProductFilterProce5sor for supplier_id fiel
E. In the custom filter, apply the needed join and filter to the passed $collection.
F. Write a before plugin On\Magento\Framework\Api\SearchCriteria\CollectionProcessorInterface: :process(). Iterate through the $searchCriteriaprovided for supplier_id, and if found, apply the needed join and filter to the passed scollection.

Answer: B

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 References: [Magento 2.4 DevDocs] [Magento Stack Exchange]
In Adobe Commerce, when you need to add a custom filter for a non-EAV attribute stored in a standalone table, the most maintainable approach is to create a custom filter for ProductFilterProcessor. This processor allows for customized search criteria handling, which can be extended to include custom joins and filters without altering core functionality or relying on plugins and observers.
? Why Custom Filter in ProductFilterProcessor is Preferred:
? uk.co.certification.simulator.questionpool.PList@205c9928
? Implementation of Custom Filter:
? Why Options A and C are Less Suitable:
:
Magento Developer Documentation onUsing Custom Filters in CollectionProcessor Magento DevDocs onDependency Injection and Virtual Types

NEW QUESTION 49
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.

Answer: A

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 References: [Magento 2.4 DevDocs] 2
The ECE-Tools package for Adobe Commerce Cloud provides a range of tools and scripts to manage and streamline deployment and maintenance tasks. Among its key features:
? Application Builds:
? uk.co.certification.simulator.questionpool.PList@2bc44d9b
? Applying Custom Patches:
? Dump Configuration for Static Content Deployment:
? Why Option A is Correct:
: Adobe Commerce Cloud documentation onECE-Tools

NEW QUESTION 50
Which price type should be used if the developer wants to provide a discount for a product based on quantity, for example, being able to buy two for X amount each?

A. Tier Price
B. Special Price
C. Group Price

Answer: A

Explanation:
Tier prices are used to provide discounts for products based on quantity. For example, you could set a tier price that allows customers to buy two products for X amount each.
The tier price is used when a developer wants to offer a discount based on the quantity purchased. For example, buying two or more units of a product at a reduced price per unit. Tier pricing allows setting different prices for different quantities, encouraging customers to buy more. Special price is a flat discounted price regardless of quantity, and group price is used to set special prices for specific customer groups, not for quantity-based discounts.

NEW QUESTION 52

Which characteristic is associated with a persistent cart?

A. By default, a persistent cookie will become inactive in 30 days.
B. While using the persistent cart, guest users do not need to log in or register to checkout
C. While the customer is logged in, If the session cookie expires, the persistent cookie will remain active

Answer: C

Explanation:
A persistent cart is a cookie that is stored on the customer's computer. This cookie allows the customer to continue shopping even if they close their browser. If the customer is logged in, the persistent cookie will remain active even if the session cookie expires.
Associated with a persistent cart in Adobe Commerce is the characteristic that while the customer is logged in, if the session cookie expires, the persistent cookie will remain active. This ensures that the customer's shopping cart is preserved even if they have been inactive and the session has expired. The persistent cookie allows the cart to be restored when the customer returns to the store.

NEW QUESTION 55
Which action, if any, should be taken to enable filtering by attribute in the category's layered navigation?

A. Set the category's "Anchor' display setting to "yes".
B. Select "With layered navigation" from the category's display mode
C. Filtering by the attribute is enabled for every category automatically.

Answer: A

Explanation:
To enable filtering by attribute in a category's layered navigation, you should set the category's "Is Anchor" setting to "Yes". This setting is found in the category's display settings within the admin panel. When a category is set as "Anchor", Magento will automatically include filtering options in the layered navigation block for all attributes that
are configured to be used in layered navigation.

NEW QUESTION 58
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

Answer: D

Explanation:
To properly configure a Magento CLI command that includes a required argument, such as --order_id, which is mandatory for processing an order, the best approach is to use the addArgument method within the configure function. This method defines required arguments for the command, ensuring the user provides the necessary data.
Option D is correct for the following reasons:
? Using addArgument for Required Inputs:The addArgument method is used here to declare order_id as a required argument. This is more appropriate than addOption
when the parameter is essential for command execution and should not be omitted. By specifying InputArgument::REQUIRED, the command ensures that the order_id must be provided by the user.
? uk.co.certification.simulator.questionpool.PList@3259569d
: According to Magento??s official developer documentation, addArgument is used for required command-line arguments, and this is standard practice for defining necessary inputs in CLI commands.
Properly Configured Command Name and Description:The setName and setDescription methods are correctly used in this option to specify the command??s name and its purpose. This helps in making the command self-descriptive, improving usability and readability for other developers or administrators using the CLI.
Options A, B, and C are incorrect because they either:
Use addOption instead of addArgument, which is less suitable for mandatory parameters (Option B).
Define the same parameter redundantly (Option C).
Use other non-standard configurations that do not align with Magento's best practices for required CLI parameters (Option A).

NEW QUESTION 63
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.xsd">
    <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

Answer: A

Explanation:
To allow theis_delayedEAV attribute to be used in the .phtml cart page, the developer needs to create a new file calledetc/catalog_attributes.xmi. This file will contain the definition of theis_delayedattribute.
The following code shows how to create theetc/catalog_attributes.xmifile: XML
<?xml version="1.0"?>
<catalog_attributes>
<attribute code="is_delayed" type="int">
<label>Is Delayed</label>
<note>This attribute indicates whether the product is delayed.</note>
<sort_order>10</sort_order>
<required>false</required>

</attribute>
</catalog_attributes>
Once theetc/catalog_attributes.xmifile has been created, theis_delayedattribute will be available in the .phtml cart page. The attribute can be accessed using thegetIsDelayed()method of theProductclass.
PHP
$product = $block->getProduct();
$isDelayed = $product->getIsDelayed();
TheisDelayedvariable will contain the value of theis_delayedattribute. 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.

NEW QUESTION 65
How can a developer override a core class method in Adobe Commerce?

A. <preference for='Magento\Catalog\Block\Product" type="Vendor\Module\Block\Producf/>
B. <typename="Magento\Catalog\Block\Product"> q <rewrite class="Vendor\Module\Block\Producf /> </type>
C. <typename="Magento\Catalog\Block\Product*> <arguments> q <argument name="rewrtte" xsi:type="object">Vendor\Module\Block\Product</argument>
</arguments></type>

Answer: A

Explanation:
To override a core class method in Adobe Commerce, the<preference> XML node is used in thedi.xmlfile of a custom module. This node specifies that, for a given interface or class, Magento should use a different class (specified in the "type" attribute) whenever the original class is requested. This allows developers to extend or modify the functionality of core Magento components by substituting their own implementations in a way that is respectful of Magento's extension mechanisms.

NEW QUESTION 67
The di. xml file of a module attaches two plugins for the class Action.
The PluginA has the methods: beforeDispatch, aroundDispatch, afterDispatch. The PluginB has the methods: beforeDispatch, afterDispatch.

```
<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

Answer: B

Explanation:
https://developer.adobe.com/commerce/php/development/components/plugins/#scenario-b-without-a-callable-around

NEW QUESTION 72
How would a developer add a sensitive environment-specific configuration value on an Adobe Commerce Cloud project?

A. Add sensitive Environment-specific variable in the Project Web Interface.
B. Connect to the server using SSH and add the configuration in the app/etc/config.php file.
C. Use the Cloud CLI for Commerce command Mgento-cloud config:set to add the configuration

Answer: A

Explanation:
To add a sensitive environment-specific configuration value on an Adobe Commerce Cloud project, the developer should use the Project Web Interface. This interface allows for the secure entry of sensitive data, which is then encrypted and stored securely. This method ensures that sensitive information is not exposed in the codebase or version control.

NEW QUESTION 76
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 <project-id> -e <environment-id> enable_smtp true
C. Access the Project Web Interface and select the Staging environmen
D. Select Configure environment.Toggle Outgoing emails On

Answer: C

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 References: [Magento 2.4 DevDocs] 1
In Adobe Commerce Cloud, email functionality for Staging and Production environments can be controlled through the Project Web Interface. To enable outgoing emails, you can toggle the setting from within the environment configuration.
? Using the Project Web Interface:
? uk.co.certification.simulator.questionpool.PList@429d3ea9
? Why Option C is Correct:
: Adobe Commerce Cloud documentation onManaging Email Settings

NEW QUESTION 81
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($file->getExtexsion() == 'pdf') {
        throw new InvalidFileException(__('The file must be PDF.'));
    }
    if($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

Answer: C

Explanation:
To add file upload validation for a custom form field in the Adobe Commerce admin panel, which should restrict the file type to .pdf and limit the file size to 2 MB, the recommended approach is to include the validation parameters directly within the
<settings> node in the form??sXML configuration. This ensures that the validation occurs on the client-side as well as server-side, providing immediate feedback to users before submission.
Option C is correct for the following reasons:
? Adding Validation Inside <settings>:By placing the <allowedExtensions> and
<maxFileSize> tags within the <settings> node of the XML configuration, the system will enforce these restrictions directly within the form component. This method

leverages Magento??s built-in support for validation settings, which ensures that only files matching the specified criteria (.pdf files of 2 MB or smaller) are accepted by the form.
? uk.co.certification.simulator.questionpool.PList@78475b0d
: Magento??s documentation on UI components highlights how to enforce file type and size restrictions through XML configurations within <settings>, making it the standard and most efficient solution for this task.
Alternatives and Limitations:
Option A: Adding validation within the controller (UploadPdf) can provide additional server- side validation, but this does not prevent the user from selecting invalid files in the first place. Server-side validation alone lacks the user experience enhancement provided by client-side feedback.
Option B: Configuring a virtual type in di.xml for validation (e.g., setting allowedExtensions and maxFileSize within a custom uploader model) is effective for backend processing but is
not as straightforward for direct form validation within the admin UI. It complicates the implementation by requiring custom backend logic where native XML validation can suffice.
Option C provides a straightforward, maintainable, and user-friendly way to implement file validation directly in the form configuration. It reduces the need for custom controller or model logic and leverages Magento??s built-in form handling capabilities.

NEW QUESTION 82
Which condition must be satisfied to ensure that a Discard Subsequent Rules option that is set to "Yes" actually prevents multiple discounts from being applied to the same product?

A. Each pricing rule must have From and To date.
B. Each pricing rule must have the defined priority.
C. Each pricing rule must be created with Coupon code

Answer: B

Explanation:
The Discard Subsequent Rules option is only applied if the pricing rules have different priorities. If two pricing rules have the same priority, the discount from both rules will be applied.
For the "Discard Subsequent Rules" option set to "Yes" to work effectively, each pricing rule must have a defined priority. When multiple discount rules can apply to the same product, Magento evaluates them in the order of their priority values. If a rule with "Discard Subsequent Rules" set to "Yes" is applied, any subsequent rules with lower priority (higher priority number) will not be applied to that product.

NEW QUESTION 85
A product has been added to the Adobe Commerce Store, and it contains a value for the custom product attribute. A merchant reports that the attribute value is not displayed in the Additional Information tab on the product detail page.
Which action will correct this problem?

A. The attribute must be moved to the specific group in the attribute set
B. The attribute property "Use in Product Tab' must be set to "yes"
C. The attribute property "Visible on Catalog Pages on Storefront" must be set to "yes".

Answer: C

Explanation:
The "Visible on Catalog Pages on Storefront" attribute property determines whether or not the attribute value is displayed in the Additional Information tab on the product detail page. If this property is set to "no", the attribute value will not be displayed.
For a custom product attribute to be displayed in the Additional Information tab on the product detail page in Magento, it needs to be visible on the catalog pages on the storefront. This visibility is controlled by the attribute property "Visible on Catalog Pages on Storefront". When this property is set to "yes", Magento includes the attribute in the Additional Information tab, making it visible to customers browsing the product. This setting ensures that only relevant and intended attributes are shown on the storefront, allowing for better product information management and customer experience.

NEW QUESTION 89
An Adobe Commerce Cloud developer wants to check the staging environment deployments history (i.e. branch, git, merge, sync). Where can the developer look up the history of the staging environment?

A. Project Web Interface
B. New Relic
C. Adobe Commerce admin panel

Answer: A

Explanation:
The Project Web Interface is the main dashboard for managing Adobe Commerce Cloud projects. This includes the ability to check the staging environment deployments history.
The developer can look up the history of deployments to the staging environment, including branch, git merge, and sync operations, in the Project Web Interface. This interface provides adetailed log of all actions taken on the project, including deployments, enabling developers to track changes and troubleshoot issues that may arise.

NEW QUESTION 91
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.

Answer: C

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 References: [Magento 2.4 DevDocs] 1
Enhanced Integration Environments in Adobe Commerce Cloud have a limit on the number of active branches. If both integration branches are currently active, one must be deactivated to create a new active branch for testing.
? Creating a New Active Integration Branch:
? uk.co.certification.simulator.questionpool.PList@400543f4
? Why Option C is Correct:
: Adobe Commerce Cloud documentation onBranch Management

NEW QUESTION 96
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.

Answer: A

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 andstores 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 References: [Magento 2.4 DevDocs]

NEW QUESTION 100
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.

Answer: C

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 References: [Magento 2.4 DevDocs]

NEW QUESTION 102
Which property allows multiple cron jobs to share the same configuration?

A. name
B. group
C. schedule

Answer: B

Explanation:
In Magento, thegroupelement in the cron job configuration allows multiple cron jobs to share the same configuration settings, such as schedule, status, and execution time limits. This grouping facilitates the management of cron jobs, making it easier to configure and maintain them, especially when multiple tasks require similar or identical settings. By assigning cron jobs to a specific group, they inherit the group's configuration, streamlining the setup process and ensuring consistent execution parameters across
related tasks.

NEW QUESTION 105
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.

Answer: B

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 References: [Magento 2.4 DevDocs]

NEW QUESTION 107
Which Cloud CLI for Commerce command can be used to quickly view a specific log file for an Adobe Commerce Cloud project?

A. magento-cloud log
B. magento-cloud logs:list
C. magento-cloud logs:show

Answer: C

Explanation:
The Cloud CLI for Commerce command that can be used to quickly view a specific log file for an Adobe Commerce Cloud project ismagento-cloud logs:show. This command allows developers to view log files directly from the command line, which is useful for debugging and monitoring the application's state without needing to access the file system directly.

NEW QUESTION 109
How would a developer access RabbitMQ data on an Adobe Commerce Cloud Production environment?

A. Using Project Web Interface
B. Using local port forwarding
C. Using RabbitMyAdmin

Answer: B

Explanation:
To access RabbitMQ data on an Adobe Commerce Cloud Production environment, you can use local port forwarding. This allows you to forward a port on your local machine to a port on the Production environment. This way, you can connect to RabbitMQ from your local machine.
A developer would access RabbitMQ data on an Adobe Commerce Cloud Production environment using local port forwarding. This is done via an SSH tunnel that securely forwards a port from the local machine to the RabbitMQ service on the cloud environment. RabbitMyAdmin (an option that does not exist) and the Project Web Interface do not provide direct access to RabbitMQ data.

NEW QUESTION 113
Which command can be used to display a full list of enabled and disabled Magento modules?

A. bin/magento module:all
B. bin/magento modulestatus
C. bin/magento module:show

Answer: B

Explanation:
The command to display a full list of enabled and disabled Magento modules isbin/magento module:status. This command provides a comprehensive overview of all modules within the Magento instance, categorizing them into enabled and disabled modules. This information is crucial for debugging and development purposes, as it allows developers to quickly understand which components of Magento are active and which are not, facilitating troubleshooting and configuration adjustments.

NEW QUESTION 116
How would a developer enable the magnification of CSS files on an Adobe Commerce Cloud Staging environment?

A. Locally from the command linebin/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 environmen
D. From the command line

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

Answer: C

Explanation:
For Adobe Commerce Cloud environments, modifying configuration settings often involves using the ece-tools command-line interface. To enable CSS minification on a staging environment, SSH into the environment and use ece-tools to set the configuration.
? Using ece-tools for Cloud Environments:
? uk.co.certification.simulator.questionpool.PList@79fa8c23
? Why Option C is Correct:
? Steps:
: Adobe Commerce Cloud documentation onUsing ece-tools

NEW QUESTION 120
Which two attribute input types can be used for a date? (Choose two.)

A. Timezone
B. Schedule
C. Date and Time

D. Date

Answer: CD

Explanation:
The two attribute input types that can be used for a date are Date and Time and Date. These input types allow the user to select a date or a date and time from a calendar widget.
The Timezone and Schedule input types do not exist in Adobe Commerce. Verified References: [Adobe Commerce User Guide - Create a product attribute]
In Magento, attribute input types define the type of data an attribute can hold and how it should be inputted or displayed in the UI. For dates, Magento provides specific input types to handle date-related data efficiently. The "Date" input type is used for attributes that require only a date, without a time component, suitable for birthdays, anniversaries, or any date-specific information. The "Date and Time" input type, on the other hand, includes both date and time components, ideal for events, promotions, or any scenario where the time of day is relevant. These input types ensure data consistency and provide a user-friendly interface for selecting dates and times.

NEW QUESTION 125
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 actio
C. They should implement this method: all non-validated POST datagets stripped out of the request and an error is thrown.
D. Form key validation runs on all non-AJAX POST requests, the developer needs to add the for_key to their requests.

Answer: C

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 References:https://magento.stackexchange.com/questions/95171/magento-2-form- validation
In Adobe Commerce, when handling POST requests from forms on the frontend, form key validation is enabled by default as a security measure to prevent Cross-Site Request Forgery (CSRF) attacks. This validation checks that the form submission is coming from the same origin by including a unique token (form key) in the request. If this form key is missing or incorrect, the request will fail, and an error message may be shown on the frontend.
In this scenario:
? Since the developer has used \Magento\Framework\App\Action\HttpPostActionInterface, which is appropriate for handling POST requests, it??s likely that the error they encounter is due to missing form key validation.
? The solution is to ensure that the form includes a hidden input field for the form key. Adobe Commerce automatically adds this key in forms if you use the \Magento\Framework\Data\Form\FormKey model to get the form key value. To implement this:
? Ensure the form includes the form key:
<input name="form_key" type="hidden" value="<?= $block->escapeHtml($block->getFormKey()) ?>" />
? The form key should also be included in the POST data sent to the controller. If it??s missing, Adobe Commerce will not process the request.
Additional Resources:
? Adobe Commerce Developer Guide: Form Key
? Magento 2.4 Form Key and CSRF Protection

NEW QUESTION 129
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> <parent-environment-id>?

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.

Answer: C

Explanation:
The type of environment that will be provisioned when launching the CLI for Commerce command magento-cloud environment:branch <environment-name> <parent- environment-id> 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 References: [Magento 2.4 DevDocs]

NEW QUESTION 134
A developer has informed the Adobe Support team about a planned traffic surge on an Adobe Commerce Cloud project that will take place in a little over 48 hours. What is an advantage of this prior notice?

A. When the traffic arrives, extra server resources will be available.
B. The project will temporarily use an upgraded Fastly plan
C. The Support team will monitor the website during that time

Answer: C

Explanation:
Informing the Adobe Support team about a planned traffic surge allows them to monitor the website during that time. With prior notice, the support team can

ensure that they are prepared to quickly respond to any issues that arise due to the surge. While extra server resources or an upgraded Fastly plan may be possible outcomes, the primary advantage of advance notice is proactive monitoring and support during expected high traffic events.

NEW QUESTION 137
Which method type can be intercepted by plugins?

A. final
B. static
C. public

Answer: C

Explanation:
In Magento, plugins (Interceptors) can only intercept public methods. This is because the plugin system relies on Magento's object manager to dynamically create proxy classes that can intercept method calls. Since private and final methods are not accessible from outside the class they are defined in, and static methods are not called on an object instance, these method types cannot be intercepted. This mechanism allows for the extension and customization of Magento's core behavior in a transparent and non-intrusive manner.

NEW QUESTION 141
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.

Answer: B

Explanation:
In Adobe Commerce Cloud, outgoing emails are disabled by default on Integration environments to prevent test or development emails from being sent to real customers.
? Email Configuration on Integration Environments:
? uk.co.certification.simulator.questionpool.PList@773ee80d
? Why Option B is Correct:
: Adobe Commerce Cloud documentation onEmail Configuration

NEW QUESTION 145
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.

Answer: C

Explanation:
https://developer.adobe.com/commerce/php/best- practices/extensions/observers/#do-not-rely-on-invocation-order

NEW QUESTION 148
A merchant wants to include taxes In an Adobe Commerce store. Which option can have a tax class assigned to it?

A. Order
B. Shipping
C. Category

Answer: B

Explanation:
According to the Adobe Commerce User Guide, a tax class can be assigned to either a product or a customer group in Adobe Commerce. A product tax class determines how a product is taxed, while a customer tax class determines how a customer is taxed based on their location and group membership. Shipping is considered as a product tax class in Adobe Commerce, and it can be assigned to different shipping methods or rates. The other options are not valid for assigning a tax class.
In Adobe Commerce, tax classes can be assigned to products and shipping. Categories, however, do not have tax classes assigned to them directly. Tax classes applied to shipping allow merchants to specify how taxes should be calculated for shipping costs, making option B the correct answer. Orders and categories do not have direct associations with tax classes in the same way products and shipping do.

NEW QUESTION 149
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.

Answer: B

NEW QUESTION 152
A developer found a bug inside a private method of a third party module class. How can the
developer override the method?

A. Create a custom class with corrected logic, and define the class as preference in the preferences.xml.
B. Create a custom class with the corrected logic, and define the class as a preference for original one in the di xml.
C. Create a plugin, implement correct logic in the after" method, and then define the plugin in the di.xml.

Answer: B

Explanation:
To override a private method in a third-party module class, the most effective approach is to use a preference. This involves creating a custom class with the corrected logic and then defining this class as a preference for the original one in thedi.xmlfile. Plugins cannot be used to override private methods, final methods, final classes, or classes created without dependency injection. Therefore, the preference mechanism, which allows for the substitution of the entire class, becomes the viable method to override a private method and modify its behavior.

NEW QUESTION 156
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.

Answer: C

Explanation:
Theinput_typeattribute 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. Thetextareainput type is used for WYSIWYG fields. If theinput_typeattribute is not set totextarea, then the attribute will not appear as WYSIWYG in the admin panel.
To fix this, the developer should set theinput_typeattribute totextarea.

NEW QUESTION 159
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.

Answer: B

Explanation:
To create a snapshot before deploying changes in Adobe Commerce Cloud (Pro Plan), the recommended approach is to use the Cloud CLI, which provides a dedicated command for creating snapshots. This allows for quick rollback if any issues arise post-deployment.
? Creating a Snapshot with Cloud CLI:
? uk.co.certification.simulator.questionpool.PList@474cafe0
? Why Option B is Correct:
: Adobe Commerce Cloud documentation onCreating and Managing Snapshots https://experienceleague.adobe.com/docs/commerce-cloud-service/user-guide/architecture/pro-architecture.html?lang=en#backup-and-disaster-recovery

NEW QUESTION 161
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

Answer: A

Explanation:
The Content Security Policy (CSP) in Adobe Commerce (Magento) restricts the types of content that can be loaded on a page to protect against malicious attacks, such as cross-site scripting (XSS). When an iframe is added, and a JavaScript library is loaded from an external source, these resources must be whitelisted explicitly using the csp_whitelist.xml file.
In this specific case:
? The frame-src directive controls the sources from which iframes can be embedded. Since the developer is embedding an iframe from an external domain, they need to whitelist this domain for frame-src.
? The script-src directive controls the sources from which JavaScript files can be loaded. The external JavaScript library must be whitelisted under script-src to allow it to execute.
Therefore, the correct policy IDs to whitelist are:
? frame-src: to allow the embedding of content from an external domain in an iframe.
? script-src: to allow the loading and execution of JavaScript files from the external domain.
Here??s how to update the csp_whitelist.xml file with the correct directives:
<?xml version="1.0"?>
<whitelist xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="urn:magento:module:Magento_Csp:etc/csp_whitelist.xsd">
<policy id="frame-src">
<values>
<value id="your-external-domain.com"/>
</values>
</policy>
<policy id="script-src">
<values>
<value id="your-external-domain.com"/>
</values>
</policy>
</whitelist>
Replace your-external-domain.com with the actual domain of the external iframe and JavaScript source.
Additional Resources:
? Adobe Commerce Developer Guide: Content Security Policy (CSP)
? CSP Policies and Directives: Explanation of all supported CSP directives and how to configure them.

NEW QUESTION 164
A new custom module is built for the existing Adobe Commerce store. A merchant has requested a few frontend updates. For this, a developer has to implement a custom style.
What is the location of the less file that will be included by default?

A. view/{area}/web/css/style less
B. view/{area}/web/css/source/main less
C. view/{area}/web/css/source/_module.less

Answer: B

Explanation:
Theview/{area}/web/css/source/main.lessfile is the default less file that is included by default. This file contains the main styles for the module.
In a custom module in Adobe Commerce, the default location for including custom LESS styles is typicallyview/{area}/web/css/source/_module.less. However, the most commonly used file for adding global styles isview/{area}/web/css/source/_extend.less. Theview/{area}/web/css/style.lessfile is not standard, and themain.lessfile is used as an entry point for compilation but typically does not contain custom styles directly.

NEW QUESTION 165
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 <prefffrence> 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 CartItemPrices {
    base_price: Money!
}
```

A black text on a white background AI-generated content may be incorrect.
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>
```

A screen shot of a computer code
AI-generated content may be incorrect.

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

Answer: B

Explanation:
The developer can extend the existing CartltemPrices GraphQL schema to include a custom base_price field by adding the following code to the module??s etc/schema.graphqls file:
extend type CartltemPrices { base_price: Money! @doc(description: ??The base price of the cart item??) }
This code adds a new field called base_price to the CartltemPrices 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 References: [Magento 2.4 DevDocs] [Magento Stack Exchange]

NEW QUESTION 166
What is the correct way to inject CMS block in a layout?

A. <block class="Magento\Cms\Block\Block" name="blockjdentifier"> <arguments>
q<argumentname="block_id"xsi:type="string">my_cms_block_identifier</argument></arguments> </block>
B. <block class="Magento\Cms\Block\Block" name="block_identifier"> q<actionmethod="setBlock'>my_cms_block_identifier</action> </block>
C. <referenceBlock name="content"> <block class="Magento\Cms\Block\Block" name="block_identifier' identifier="my_cms_block_ldentrfier" /> </referenceBlock>

Answer: A

Explanation:
The correct way to inject a CMS block into a layout in Adobe Commerce is by using the<block>element with the classMagento\Cms\Block\Blockand specifying the block identifier through an<argument>element with the name "block_id". This is shown in option A. The<block>tag defines the block class and name, and the<arguments>tag contains child<argument>tags for each argument, where the "block_id" argument specifies the identifier of the CMS block to be injected.

NEW QUESTION 170
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.

Answer: C

Explanation:
The developer can resolve this issue by creating a plugin for theMagento\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 theexecuteRow()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 theadjustmentdata from the product to the product price index. Once the product price index is updated, the layered navigation will display the correct price.

NEW QUESTION 171
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 $lock parameter, i.
B. setLocale($newLocaleCode,true), to override and lock the locale of the emulated stor
C. If this is set and locked initially then the environment emulation will not be able to override this.
D. 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.
E. 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.

Answer: B

Explanation:
ThestartEnvironmentEmulation()method resets the translation locale to the one of the emulated stores, which overrides the locale the developer has set when the order ofsetLocale()andstartEnvironmentEmulation()is used as displayed above.
The correct way to achieve the desired result is to use theemulate()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. ThestartEnvironmentEmulation()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.
Theemulate()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.

NEW QUESTION 172
A developer is creating a class \Vendor\Module\Model\MyModeL How should that class be defined as transient in di.xml?
A. <type name="\Vendor\Module\Model\MyModer transient="true7>
B. <type name="\Vendor\Module\Model\MyModer singleton='false7>
C. <type name='\Vendor\Module\Model\MyModer shared="false7>

Answer: C

Explanation:
To define a class as transient in Magento'sdi.xml, the correct approach is to set thesharedattribute to "false" for that class. This tells Magento's object manager not to use the singleton pattern for this class, meaning a new instance will be created each time the class is requested. This is particularly useful for classes that should not maintain state across different areas of the application or during a single request lifecycle.

NEW QUESTION 175
A developer needs to extend the existing jQuery widget. Which jQuery function is used for this purpose?

A. $.mage
B. $.ui
C. $.widget

Answer: C

Explanation:
To extend an existing jQuery widget in Adobe Commerce, the$.widgetfunction is used. This function is part of jQuery UI's widget factory and is a powerful tool for creating stateful plugins with a consistent API. It allows developers to create a new widget that inherits from an existing widget, enhancing or modifying its functionality, making option C the correct answer.

NEW QUESTION 178
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 Merge
B. 1. Checkout to Production environment* 2. Use the magento-cloud synchronize <environment-ID> Commerce CLI Command
C. 1, Log in to the Project Web Interface.* 2. Choose the Staging environment, and click Sync

Answer: C

Explanation:
To update the Staging environment with the latest code from the Production
environment on an Adobe Commerce Cloud project, the developer would log in to the Project Web Interface, choose the Staging environment, and then click Sync. This action synchronizes the environments, bringing the latest changes from Production into Staging.

NEW QUESTION 179
What does a URL Rewrite do?

A. It updates the URL that is stored on the server.
B. It changes the way a URL appears in the browser
C. It updates the URL to a domain that is not being Indexed.

Answer: B

Explanation:
A URL Rewrite in Magento changes the way a URL appears in the browser. This is particularly useful for improving the readability and SEO of a URL. For example, a URL rewrite can be used to transform a long and complex URL into a shorter and more user-friendly version. It's important to note that while a URL rewrite changes the URL's appearance in the browser, it doesn't change the actual location of the resource on the server. This distinction is crucial for understanding how Magento handles URL rewrites and redirects, facilitating a more intuitive navigation structure within the store without altering the underlying server resources.

NEW QUESTION 180
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

Answer: A

Explanation:
Upgrading to Enhanced Integration Environments in Adobe Commerce Cloud requires specific steps to ensure that the environment is prepared for the upgrade, which includes managing integration branch limits and coordinating with Adobe support.
? Limiting Integration Branches:
? uk.co.certification.simulator.questionpool.PList@6a215712
? Submitting a Support Ticket:
? Why Option A is Correct:
: Adobe Commerce Cloud documentation onEnhanced Integration Environments

NEW QUESTION 182
.....

Thank You for Trying Our Product

We offer two products:

1st - We have Practice Tests Software with Actual Exam Questions

2nd - Questons and Answers in PDF Format

AD0-E724 Practice Exam Features:

* AD0-E724 Questions and Answers Updated Frequently

* AD0-E724 Practice Questions Verified by Expert Senior Certified Staff

* AD0-E724 Most Realistic Questions that Guarantee you a Pass on Your FirstTry

* AD0-E724 Practice Test Questions in Multiple Choice Formats and Updatesfor 1 Year