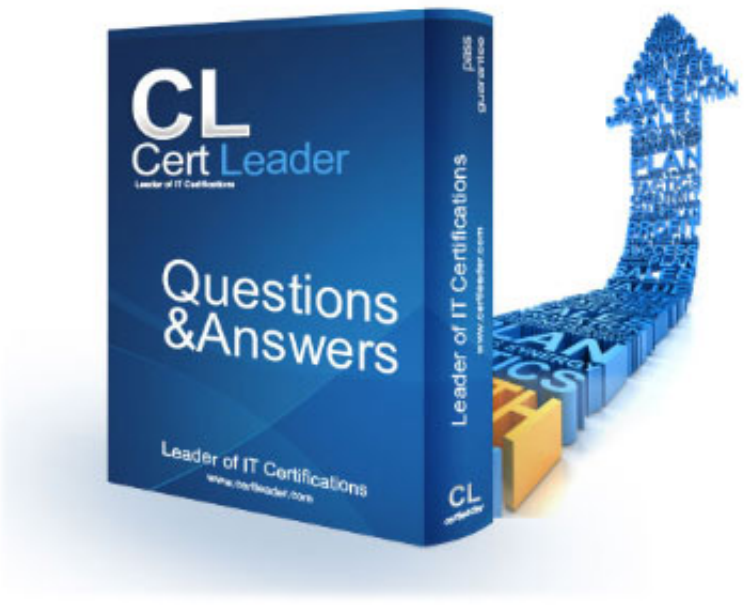
AD0-E724 Dumps

Adobe Commerce Developer Professional

https://www.certleader.com/AD0-E724-dumps.html



NEW QUESTION 1
An Adobe Commerce developer is creating a module (Vendor.ModuleName) to be sold on the Marketplace. The new module creates a database table using declarative schema and now the developer needs to make sure the table is removed when the module is disabled.
What must the developer do to accomplish this?

A. There is nothing further the developer needs to d
B. The table will be removed when the module is disabled and bin/magento setup:upgrade is run.
C. There is nothing further the developer needs to d
D. The table will be removed when the when bin/magento module:uninstall vendor_ModuleName is run.
E. Add a schema patch that implements Magento\Framework\setup\Patch\PatchRevertabieinterface and drops the table in the revert function.

Answer: A

Explanation:
When you disable a module, Magento removes all of its declarative schema changes from the database the next time you run bin/magento setup:upgrade."
This means that when the developer disables the module and runs setup:upgrade, Adobe Commerce will automatically handle the removal of the database table created by the module's declarative schema.
For reference, here are some key points from the documentation:
? [Disable a Module](https://x.com/i/grok?text=Disable%20a%20Module)- This section explains how Magento handles the database schema when a module is disabled.
? Declarative Schema Configuration- Provides an overview of how declarative
schema works, including its behavior when modules are disabled or uninstalled. Therefore, based on the official Adobe Commerce documentation, the correct action for the
developer is to do nothing further beyond disabling the module and running bin/magento setup:upgrade. Magentowill take care of removing the table associated with the module as part of its schema management process.

NEW QUESTION 2
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 3
A developer would like to initialize a theme in Adobe Commerce. Which two files are required to complete this task? (Choose two.)

A. theme.less
B. registration.php
C. composerjson
D. themexml

Answer: BC

Explanation:
To initialize a theme in Adobe Commerce, at least two files are required: registration.phpandtheme.xml. Theregistration.phpfile is used to register the theme within the system, andtheme.xmldefines the theme's name, parent (if any), and other metadata. Thetheme.lessfile is not required for theme initialization but may be used for custom styling. The correct option fortheme.xmlis represented as "theme.xml" (D), not "themexml" as mentioned in the options.

NEW QUESTION 4
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 5
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 6
An Adobe Commerce developer was asked to provide additional information on a quote. When getting several quotes, the extension attributes are returned, however, when getting a single quote it fails to be returned.
What is one reason the extension attributes are missing?

A. The developer neglected to add coiiection="trueM to their attribute in etc/extension_attributes.xmi file.ottribute code="my_attributesM type="MyVendor\MyModule\Api\Data\^AttributeInterface[]M collection="true" />
B. The developer neglected to provide a plugin On Hagento\Quote\Api\CartRepositoryInterface: :get.

C. The developer neglected to implement an observer on the coiiection_ioad_after event.

Answer: B

Explanation:
In Adobe Commerce, to retrieve custom extension attributes on an entity such as a quote, you need to ensure that these attributes are properly loaded when accessing the entity directly via repository interfaces. When fetching a single quote, it is essential to use a plugin on CartRepositoryInterface::get to include the extension attributes as this method is responsible for loading individual quote data.
If the plugin is missing for the get method, the extension attributes won??t be loaded for single quote retrieval, leading to their absence in the result.
Additional Resources:
? Adobe Commerce Developer Guide: Extension Attributes
? Magento 2 Repositories and Plugins

NEW QUESTION 7
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 8
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 9
In a new release of a module, a developer decides to rename a table that was defined in the earlier versions. Which action, if any, allows the developer to prevent data loss?

A. Define onCreate="migrateDataFromAnotherTable(old_table_name)" attribute in the table tag.
B. Declarative schema supports RENAME TABLE', so the data will be migrated to the new table automatically.
C. Define the table and columns mapping in the db.schema_whitelist.json

Answer: C

Explanation:
When renaming a table in Magento, to prevent data loss, the developer must define the table and its columns mapping in thedb_schema_whitelist.jsonfile. This declarative schema approach ensures that the data migration tool knows about the changes and can migrate data from the old table to the newly named table without losing any data.

NEW QUESTION 10
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 10
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 12
What is the command used to upgrade ECE-Tools on an Adobe Commerce Cloud platform?

A. php ./vendor/bin/ece-tools upgrade
B. composer update magento/ece-tools --with-all-dependencies
C. magento-cloud ece-tools:upgrade

Answer: B

Explanation:
To upgrade ece-tools on Adobe Commerce Cloud, the recommended command is to use Composer to ensure that all dependencies are resolved and updated

accordingly. The command composer update magento/ece-tools --with-all-dependencies will update ece-tools along with any other dependencies required.
? Using Composer for Dependency Management:
? uk.co.certification.simulator.questionpool.PList@249e6cc2
? Why Option B is Correct:
: Adobe Commerce Cloud documentation onUpgrading ECE Tools

NEW QUESTION 15
Which type of product would assist a seller who would like to offer an electronic version of an album for sale and sell each song individually?

A. Downloadable
B. Simple
C. Configurable

Answer: A

Explanation:
The type of product that would assist a seller in offering an electronic version of an album and selling each song individually is a Downloadable product. In Adobe Commerce, a Downloadable product type is specifically designed for selling digital products such as music files, which customers can download. This type of product allows sellers to upload individual songs and sell them separately or as part of a complete album.

NEW QUESTION 19
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 23
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 27
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 31
Which action, if any, should be taken to forbid Adobe Commerce Admin from performing specific actions?

A. Create a new user role with custom-defined resources, and assign it to the admin user
B. This action cannot be taken since all admin users must have full access.
C. Enable custom roles in the store configuration, and assign admin user ID(s).

Answer: A

Explanation:
To forbid Adobe Commerce Admin from performing specific actions, a developer should create a new user role with custom-defined resources, and assign it to the admin user. This can be done by going toSystem > Permissions > Rolesand creating a new role. In theResourcessection, the developer can select the specific resources that they want to restrict the admin user from accessing.
To restrict specific actions within the Adobe Commerce Admin, the recommended approach is to utilize Magento's Access Control List (ACL). This can be done by creating a new user role with custom-defined resources and assigning this role to the admin user. This approach allows for granular control over what actions an admin user can perform by specifying allowed resources within the role. Magento's ACL system is designed to manage permissions effectively, ensuring that users only have access to the necessary functionalities required for their role.

NEW QUESTION 35
On the Adobe Commerce Cloud Project Web Interface, what will be performed when clicking on the "Delete" button of an integration environment?

A. The environment is marked as "inactive", the git branch is still present but the database is deleted.
B. The environment is completely delete
C. Including git branch and database.
D. The environment is marked as "inactive", the git branch and the database are still present.

Answer: B

Explanation:
On the Adobe Commerce Cloud Project Web Interface, clicking on the "Delete" button of an integration environment will completely delete the environment, including the associated git branch and database. This action is irreversible and is used to remove an environment that is no longer needed. The environment, once deleted, frees up resources for the project and cannot be restored.

NEW QUESTION 36
Which file on an Adobe Commerce Cloud project allows a developer to upgrade the PHP version and enable/disable a PHP extension?

A. magento.app.yaal
B. .magent
C. en
D. yaml
E. php.ini

Answer: B

Explanation:
The.magento.env.yamlfile is used on an Adobe Commerce Cloud project to customize the environment configuration, including the PHP version and enabling/disabling PHP extensions. This YAML configuration file provides the ability to manage service configurations and is essential for customizing the Cloud environment.

NEW QUESTION 40
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 44
Which two attribute input types does Magento already have by default? (Choose two.)

A. Multiple Select
B. Text Field
C. Geographic coordinate
D. Numeric Field

Answer: AB

Explanation:
The two attribute input types that Adobe Commerce already has by default are Multiple Select and Text Field. Multiple Select allows the user to select multiple values from a list of options. Text Field allows the user to enter text in a single line.
The Geographic coordinate and Numeric Field input types do not exist in Adobe Commerce.
Verified References: [Adobe Commerce User Guide - Create a product attribute] Magento, by default, provides various attribute input types to accommodate different data entry needs for product and customer attributes. The "Text Field" input type allows for
single-line text input, suitable for short, textual data such as names, titles, or any other brief information. The "Multiple Select" input type enables the selection of multiple options from a predefined list, useful for attributes with multiple applicable values like colors, sizes, or features. These input types are part of Magento's

flexible attribute system, allowing for customizable data entry fields that cater to diverse product and customer data requirements.

NEW QUESTION 47
An Adobe Commerce developer has been asked to modify the PageBuilder slider content type to allow a new custom content type (other than slide) to be assigned as a child. The developer has already created the new content type called improved_slide in their module. They now need to create a new view/adminhtml/pagebuilder/content_type/slider. xml file in their module to allow the new content type to be a child of slider content types.
What is the correct xml to accomplish this?
A)

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

B)

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

C)

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

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

Answer: C

Explanation:
The correct answer is Option C. This XML configuration is the correct way to define allowed child content types for a slider content type in Magento's PageBuilder.
Magento PageBuilder Content Type Structure:
In PageBuilder, each content type can specify which other content types are allowed as children.
This is done by defining the allowed_children array within the content type??s XML configuration.
Analyzing Option C:
Arguments Definition: Option C uses the <arguments> node to define a new argument named allowed_children.
Array Structure: This argument is an array (xsi:type="array") that includes an item specifying the improved_slide as an allowed child with xsi:type="string".
This configuration is correct because it explicitly defines which child content types are allowed for the slider content type, adhering to Magento??s structure for allowed child
elements in PageBuilder.
Why Options A and B are Incorrect:
Option A: Uses a <children> node with policy="allow", which is not the standard way to define allowed children for PageBuilder content types. This format is incorrect and won??t be recognized by PageBuilder.
Option B: Uses <allowed_descendants>, which also doesn??t align with the way Magento??s PageBuilder expects child content types to be declared. The correct term is allowed_children, not allowed_descendants.
References:
Magento PageBuilder Development Guide - This guide provides insights into customizing PageBuilder and managing content types.
Configuring Content Types in PageBuilder - Documentation on how to define allowed children for custom content types.
Adobe Commerce PageBuilder Content Types XML Reference - Details on the correct XML structure for PageBuilder content types.
Option C??s configuration aligns with Adobe Commerce PageBuilder??s structure for defining which content types can be nested within another, making it the correct choice.

NEW QUESTION 48
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 53
What is one purpose of a customer data JS library?

A. It stores the customers credit card info for usage in the checkout.
B. It stores private customer data in local storage
C. It stores the customer's username and password for easier frontend login.

Answer: B

Explanation:
The customer data JS library is used to store private customer data in local storage. This data can be used to improve the customer's experience on the store, such as by remembering their shipping address or their preferred payment method.
The customer data JS library in Adobe Commerce is used for managing customer data on the client side, such as the shopping cart, comparison list, and wishlist. It does not store sensitive information like credit card details or usernames and passwords. Instead, it utilizes local storage to keep a private data section where customer-specific data is stored securely and accessed via JavaScript, making option B correct.

NEW QUESTION 57
A seller would like to offer an electronic version of an album by selling each song individually. Which layout can be used to customize a product page layout for this item?

A. catalog_product_view_type_downloadable
B. catalog_product_view_type_configurable
C. catalog_product_view_category

Answer: A

Explanation:
Thecatalog_product_view_type_downloadablelayout can be used to customize a product page layout for a downloadable product. This layout includes the product details, the product reviews, and the download links for the product's files.
For selling electronic versions of albums with individual songs, the downloadable product
type in Adobe Commerce is appropriate. To customize the product page layout specifically for downloadable items, the layout handlecatalog_product_view_type_downloadableis used. This layout handle allows developers to target downloadable products specifically and apply custom layouts or templates, making option A correct.

NEW QUESTION 59
A client would like to add an image icon in front of the telephone field to the shipping address form on a checkout page. What is the correct way to modify the UI component to set a custom template file for the field?
A)

```
...
<item name="telephone" xsi:type="array">
<arguments name="config" xsi:type="array">
<item name="elementTmpl" xsi:type="string">%Vendor_Module%/form/element/%your_template%</item>
</arguments>
</item>
...
```

B)

```
...
<block name="telephone" xsi:type="array">
<arguments name="config" xsi:type="array">
<item name="elementTmpl" xsi:type="string">%Vendor_Module%/form/element/%your_template%</item>
</arguments>
</block>
...

...
```

C)

```
...
<block name="telephone" xsi:type="array">
<arguments name="config" xsi:type="array">
<item name="elementTmpl" xsi:type="string">%Vendor_Module%/form/element/%your_template%</item>
</arguments>
</block>
...
```

D)

```
...
<item name="telephone" xsi:type="array">
<item name="config" xsi:type="array">
<item name="elementTmpl" xsi:type="string">%Vendor_Module%/form/element/%your_template%</item>
</item>
</item>
...
```

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

Answer: B

Explanation:
To add an image icon in front of the telephone field in the shipping address form on the checkout page, the correct way is to modify the UI component by setting a custom template file for the field. The snippet in option B is the correct one because it uses the<block>element and sets theelementTmplto the custom template path within the argumentsnode under theconfignode. This modification will instruct Magento to use the specified custom template file for rendering the telephone field, allowing for the addition of an image icon in front of it.

NEW QUESTION 63
On an Adobe Commerce Cloud platform, in which order does the ECE-Tools package apply patches?

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

Answer: B

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

NEW QUESTION 66
When researching some issues with the indexer, an Adobe Commerce developer is seeing errors in the logs similar to Memory size allocated for the temporary

table is more than 20% of innodb_buffer_pool_size. It is suggested that the client update innodb_buf f er_pool_size or decrease the batch size value.
Why does decreasing the batch size value improve performance?

A. This decreases memory usage for the temporary table.
B. This allows for a longer timeout per batch process.
C. This allows for more PHP threads to be utilized during the process.

Answer: A

Explanation:
Decreasing the batch size value improves performance by reducing the memory usage for the temporary table. The batch size value determines how many rows of data are processed at a time by the indexer. A large batch size value can cause the allocated memory size for the temporary table to exceed 20% of innodb_buffer_pool_size, which can result in errors and slow down the indexing process. By lowering the batch size value, the indexer can process the data more efficiently and avoid memory issues. Verified References: [Magento 2.4 DevDocs] [Magento Stack Exchange]
The error message regarding innodb_buffer_pool_size indicates that the temporary table??s memory usage is high. Decreasing the batch size value directly reduces the number of rows processed in each batch, which in turn reduces the memory requirements for the temporary table.
? Impact of Batch Size on Memory Usage:
? uk.co.certification.simulator.questionpool.PList@391e3ecc
? Why Option A is Correct:
? Recommendations:
: Magento DevDocs onIndexer Configuration and Troubleshooting MySQL Documentation onInnoDB Buffer Pool

NEW QUESTION 67
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 72
How should a grid or form be included in an admin page layout using the Ul Component?

A. <referenceContainername='content"> q <uiComponentname="example_listing.xml7></referenceContainer>
B. <referenceContainername='contenf> q <uiComponent name="example_listing7></referenceContainer>
C. <referenceContainername='content"><uiComponentname="Vendor_Module::ul_component/example_listing.xml7></referenceContainer>

Answer: B

Explanation:
To include a grid or form in an admin page layout using the UI Component, the correct approach is to use the<uiComponent name="example_listing"/>within a <referenceContainer name='content'>block of the layout XML file. This method directly references the UI component's configuration file (e.g.,example_listing.xml) which defines the structure andfunctionality of the UI component, such as grids or forms. This configuration file is located under theview/adminhtml/ui_componentdirectory of the corresponding module.

NEW QUESTION 74
A message queue currently has queue/consumer-wait-for-messages set to true, which allows the consumer process to run until a message is inserted into the queue. A piece of functionality is driven by data stored in the model
\Magento\variable\Model\variable and this value is only loaded once during the consumer run. If the variable is updated we want the consumer to restart so that the new value is loaded into memory without having to reload the variable on each message consumed.
The Adobe Commerce developer has created an after plugin on the
\Magento\Variable\Model\variable:: save() function.
How would the developer use the plugin to trigger the consumer restart?

A. Call the function\Magento\Framework\MessageQueue\PoisonPill\Poi5onPillPutInterface::put().
B. Call the function\Magento\Framework\MessageQueue\Consumers\TriggerRe5tartInterface:trigger().
C. Set the global Cache key trigger_consumer_restart t0 1.

Answer: A

Explanation:
In Adobe Commerce, when a consumer process needs to restart, the PoisonPillPutInterface can be used. The put() method of this interface triggers a "poison pill," which signals running consumers to restart. This is particularly useful when updated data needs to be reloaded into memory, as in this scenario with the \Magento\Variable\Model\Variable. Poison Pill Mechanism:
The poison pill method tells the message queue consumer to stop its current process and restart, allowing it to pick up any configuration or data changes that occurred since the last start.
Why Option A is Correct:
By calling PoisonPillPutInterface::put(), the consumer will receive a restart signal, which is ideal for cases where data loaded at the beginning of the consumer??s lifecycle must be updated.
Option B is incorrect because TriggerRestartInterface::trigger() does not exist in the Magento framework. Option C is also incorrect as setting a cache key alone does not trigger a consumer restart.
Implementation:
Use an after plugin on the save() method to trigger PoisonPillPutInterface::put() after the variable is saved.
References:

Magento DevDocs on Message Queues and Poison Pill

NEW QUESTION 79
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 80
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 81
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 82
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 86
An Adobe Commerce Developer wishes to add an action to a pre-existing route, but does not wish to interfere with the functionality of the actions from the original route.
What must the developer do to ensure that their action works without any side effects in the original module?

A. In the route declaration, use the before or after parameters to load their module in before or after the original module.
B. Inject the new action into the standard router constructor's $actiomist parameter.
C. Add the action into to the controllers/front_name/ in My.Module, Magento will automatically detect and use it.

Answer: C

Explanation:
In Magento 2, to add a new action to a pre-existing route without interfering with the existing functionality, the new action should be placed in the same directory structure under the new module??s controller namespace. Magento's autoloading mechanism will automatically detect and include it alongside the original module's actions. Here's how you can achieve this:
? Directory Structure: Ensure that your new module's controller directory structure
mirrors that of the original module.
? Controller Action: Define the new action within the appropriate directory.
For example, if you want to add a new action to thecatalogroute inMagento_Catalog:
? Create a directory structureapp/code/My/Module/Controller/Catalog/.
? Add your new action class in this directory, for example: namespace My\Module\Controller\Catalog;
use Magento\Framework\App\Action\Action; use Magento\Framework\App\Action\Context;
class NewAction extends Action
{
public function construct(Context $context)
{
parent:: construct($context);
}
public function execute()
{
// Your custom logic here
}
}
Router Configuration: Magento automatically includes this action when the route matches. By following this method, you ensure that your new action is added seamlessly without modifying the original module or causing conflicts. Magento's router will include and recognize your action based on the directory and namespace conventions.
Sources:
? Fundamentals of Magento 2 Development documents .
? Magento 2 official developer documentation.

NEW QUESTION 91
When checking the cron logs, an Adobe Commerce developer sees that the following job occurs daily: main.INFO: Cron Dob inventory_cleanup_reservations is successfully finished. However, the inventory_reservation table in the database is not emptied. Why are there records remaining in the inventory_reservation table?

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

Answer: B

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

NEW QUESTION 92
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 93
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 97
Which command invalidates the index?

A. bin/magento indexerreindex <index_name>
B. bin/magento indexer:reset <index_name>
C. bin/magento Indexerinvalldate <lndex_name>

Answer: B

Explanation:
The commandbin/magento indexer:reset <index_name>is used to invalidate a specific index in Magento. When an index is invalidated, it flags the index as 'Invalid' in the Index Management section of the Magento Admin, indicating that the data is out of sync and needs to be updated. This command does not perform the reindexing itself but prepares the specified index for reindexing by marking it as needing an update. This is an important step in ensuring that the Magento storefront reflects the most current data.

NEW QUESTION 98
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 102
An Adobe Commerce developer is working on a custom gallery extension.
The module uses the Magento\catalog\Model\iinageUploader class for image uploading. The admin controller for custom image uploads is Vendor\CustomGallery\Controller\Adminhtml\Image\Upload.
The images need to be stored in different basePath and baseTmpPath than the default ones.
How can the default imageuploader class be extended and used without affecting the other modules that are already using it?
A)

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

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

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

B)

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

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

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

C)

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

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

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

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

Answer: A

Explanation:
By explicitly injecting the virtual type into the controller, you localize the configuration changes to only the desired functionality. This approach is efficient and maintains module independence, a key principle in Magento development.
Options B and C are incorrect for the following reasons:
Option Bdirectly modifies Magento\Catalog\Model\ImageUploader with the new paths. This change will affect all usages of the ImageUploader class across the site, which contradicts the requirement to avoid impacting other modules.
Option Cinvolves creating a virtual type and then setting it as a preference. However, using a preference would replace all instances of ImageUploader across the entire Magento application, leading to the same issue as Option B.

NEW QUESTION 106
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:
The way a developer would access RabbitMQ data on an Adobe Commerce
Cloud Production environment is by using local port forwarding. This method allows the developer to connect to the RabbitMQ service instance through an SSH tunnel and access the RabbitMQ Management UI from a web browser. The developer needs to use the magento-cloud ssh command to establish the SSH connection and the
$MAGENTO_CLOUD_RELATIONSHIPS variable to retrieve the RabbitMQ connection details and login credentials.

NEW QUESTION 110
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 115
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_Identrfier" /> </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 117
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 121
On an Adobe Commerce Cloud platform, at what level is the variable env: composer_auth located in the Project Web Interface?

A. In the Environment-specific variables.
B. In the Integration variables.
C. In the Project variables.

Answer: C

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

NEW QUESTION 122
Which hashing algorithm will Adobe Commerce choose to hash customer passwords?

A. If the Sodium extension is installed, SHA256 will be chosen, otherwise MD5 will be used as the Magento default hashing algorithm.
B. If the Sodium extension is installed, Argon 2ID13 will be chosen, otherwise SHA256 will be used as the Magento default hashing algorithm.
C. It does not matter if the Sodium extension is installed or not, the Magento hashing default algorithm will be SHA256.

Answer: B

Explanation:
If the Sodium extension is installed, Argon 2ID13 will be chosen as the Magento default hashing algorithm. Otherwise, SHA256 will be used.
The Sodium extension is a PHP extension that provides cryptographic functions. Argon 2ID13 is a password hashing algorithm that is considered to be more secure than SHA256. If the Sodium extension is installed, Magento will use Argon 2ID13 as the default hashing algorithm for customer passwords. If the Sodium extension is not installed, Magento will use SHA256 as the default hashing algorithm.
Adobe Commerce uses secure hashing algorithms for customer passwords. As of the more recent updates, Adobe Commerce defaults to using the Argon2ID13 hashing algorithm, provided that the Sodium PHP extension is available. Argon2ID is considered a secure and modern hashing algorithm designed to protect against brute-force attacks.
If the Sodium extension is not available, Adobe Commerce falls back to using SHA256, which, while secure, is not as robust as Argon2ID13.
This functionality ensures that customer data is safeguarded with the highest level of security available based on the server configuration.
Additional Resources:
? Adobe Commerce Developer Guide: Hashing Algorithms
? PHP Documentation: Argon2 and Sodium

NEW QUESTION 127
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 128
Which two methods add sorting to collections inherited from the
\Magento\Framework\Model\ResourceModel\Db\Collection\AbstractCollection class? (Choose two.)

A. setOrder
B. setSorting

C. addSorting
D. addOrder

Answer: AD

Explanation:
The two methods that add sorting to collections inherited from the \Magento\Framework\Model\ResourceModel\Db\Collection\AbstractCollection class are setOrder and addOrder. These methods allow adding one or more order clauses to a collection query.
The setSorting and addSorting methods do not exist in Adobe Commerce. Verified References: [Adobe Commerce Developer Guide - Collections]
In Magento 2, collections inherited from \Magento\Framework\Model\ResourceModel\Db\Collection\AbstractCollectionclass can be sorted using thesetOrderandaddOrdermethods. ThesetOrdermethod is used to set the order for a field in the collection, specifying the field by which to sort and the direction of the sorting (ASC or DESC). TheaddOrdermethod is similar but allows for adding multiple sorting orders to the collection, enabling more complex sorting scenarios. There are nosetSortingoraddSortingmethods in the standard Magento 2 collection classes.

NEW QUESTION 129
An integration named Marketing is created on the Adobe Commerce instance. The integration has access on Magento_Customer:: customer resources and the access token is xxxxxx.
How would the rest API be called to search the customers?

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

Answer: A

Explanation:
When using an integration token to access Magento??s REST API, you can authenticate requests by including the token in the Authorization header as a Bearer token. This allows the system to recognize the permissions assigned to the integration and grant access to the specified resources.
? Using the Access Token as Bearer Token:
? uk.co.certification.simulator.questionpool.PList@69ee4bb7
? Why Option A is Correct:
? Example Command: curl -X GET "https://magentourl/rest/V1/customers/search?searchCriteria[filterGroups][0][filters][0][field] =email&searchCriteria[filterGroups][0][filters][0][value]=example@example.com" -H "Authorization: Bearer XXXXXX"
References:
Adobe Commerce REST API documentation on Authentication Magento Integration Tokens Guide on Using Tokens

NEW QUESTION 132
Which attribute option restricts Catalog EAV attributes to only certain product types?

A. show.in
B. apply_to
C. allowed_in

Answer: B

Explanation:
Theapply_toattribute option in Magento's Catalog EAV model restricts the use of certain attributes to specific product types. By specifying product types in the apply_tofield, developers can control which attributes are applicable to which types of products, ensuring that attributes are only available where they are relevant and meaningful.

NEW QUESTION 134
What is an advantage of the read-only core file system using Adobe Commerce Cloud?

A. Ensures that all changes to the production environment are tracked.
B. Improves website performance.
C. Reduces the number of attackable surfaces significantly

Answer: A

Explanation:
The read-only core file system on Adobe Commerce Cloud ensures that all changes to the production environment are tracked. This is because any changes to the code must go through version control, and the deployment pipeline, which includes stages like build, staging, and production. This approach helps maintain consistency across environments, ensures deployment best practices, and reduces human error by preventing direct changes on production servers.

NEW QUESTION 139
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 143
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 146
.....

Thank You for Trying Our Product

* 100% Pass or Money Back

All our products come with a 90-day Money Back Guarantee.

* One year free update

You can enjoy free update one year. 24x7 online support.

* Trusted by Millions

We currently serve more than 30,000,000 customers.

* Shop Securely

All transactions are protected by VeriSign!

100% Pass Your AD0-E724 Exam with Our Prep Materials Via below:

https://www.certleader.com/AD0-E724-dumps.html