

Adobe

Exam Questions AD0-E724

Adobe Commerce Developer Professional



NEW QUESTION 1

What is the default store ID for the admin panel?

- A. 1
- B. -1

Answer: A

Explanation:

In Magento, the default store ID for the admin panel is 0. This is used as a fallback mechanism where, if the current store view's ID is not 0, Magento automatically adds 0 as a fallback. This ensures that the admin panel has a unique identifier, differentiating it from the frontend store views, which typically start with IDs higher than 0. This setup is crucial for Magento's multi-store architecture, allowing for distinct configurations and behaviors between the admin and frontend contexts.

NEW QUESTION 2

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 the sortOrder property within the plugin's declaration in the di.xml file. Specifying a lower value for the sortOrder property 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 3

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 4

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 on Fastly CDN

Adobe Commerce Cloud documentation on Deployment and CI/CD

NEW QUESTION 5

Which has its own root category?

- A. Websites
- B. Stores
- C. Store Views

Answer: B

Explanation:

In Magento, each store has its own root category. The root category acts as the top-level category under which all other categories and products are organized for that particular store. This structure allows for different catalog structures across multiple stores within a Magento installation, enabling a tailored product offering and navigation experience for each store. The ability to assign a unique root category to each store is a fundamental aspect of Magento's multi-store functionality, providing the flexibility to cater to diverse market segments or branding requirements.

NEW QUESTION 6

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 the `composer.json` file 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 7

For security reasons, merchant requested to a developer to change default admin url to a unique url for every branch/environment of their Adobe Commerce Cloud project. Which CLI command would the developer use update the admin url?

- A. `ece-tools variable:update ADMIN_URL`
- B. `magento-cloud variable:set ADMIN_URL`
- C. `bin/magento adminuri:set <admin_uri>`

Answer: B

Explanation:

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

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

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 10

A developer is making customizations in the checkout, and access to the quote's 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 file Magento_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 10

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 14

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 15

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

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

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

- A. The option's uri attribute is not valid.
- B. The layout cache needs to be refreshed.
- C. The developer has to add a formkey for the new export option.

Answer: C

Explanation:

The developer has to add a formkey for the new export option because the formkey is required for security reasons. Without the formkey, the request will be rejected and redirected to the dashboard page. Verified References: [Magento 2.4 User Guide] [Magento 2.4 DevDocs]
When adding custom export options to grids in Magento, it's crucial to include a form key for actions that involve form submission. Magento relies on form keys for CSRF (Cross-Site Request Forgery) protection, so omitting the form key can lead to redirects or failed operations.

? Form Key Requirement:

? uk.co.certification.simulator.questionpool.PList@755b4fcf

? Why Option C is Correct:

? Solution:

: [Adobe Commerce documentation onForm Key and CSRF Protection](#) [Magento DevDocs onAdding Buttons to Grids](#)

NEW QUESTION 16

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 19

What action can be performed from the Cloud Project Portal (Onboarding UI) of an Adobe Commerce Cloud project?

- A. Set your developer SSH public key.
- B. Update Project and environment variables
- C. Add a Technical Admin

Answer: C

Explanation:

The Cloud Project Portal (Onboarding UI) of an Adobe Commerce Cloud project is a web interface that allows you to perform various actions related to your project, such as creating and managing environments, deploying code, configuring services, and adding users¹. One of the actions that you can perform from the Cloud Project Portal is adding a Technical Admin, which is a user role that has full access to all environments and canperform any action on the project². To add a Technical Admin from the Cloud Project Portal, you need to follow these steps²:

? Log in to the Cloud Project Portal with your Magento account credentials.

? Click on the Users tab on the left sidebar.

? Click on the Add User button on the top right corner.

? Enter the email address of the user you want to add as a Technical Admin.

? Select the Technical Admin role from the Role dropdown menu.

? Click on the Send Invitation button.

The user will receive an email invitation to join your project as a Technical Admin. They will need to accept the invitation and set up their account before they can access your project².

NEW QUESTION 20

A developer is working on an Adobe Commerce Cloud project and wants to get connection data for the environment's deployed services. The developer has all of the necessary permissions to do this.

Which two options would the developer take to get the connection credentials? (Choose Two.)

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

Answer: AB

Explanation:

In Adobe Commerce Cloud, connection data for deployed services (such as databases, caches, and other services) can be retrieved using different methods depending on the developer's access and the tools available.

? Using magento-cloud relationships Command:

? uk.co.certification.simulator.questionpool.PList@73a02881

? Project Web Interface:

? Why Options A and B are Correct:

:

: [Adobe Commerce documentation onAccessing Services](#)

NEW QUESTION 24

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

- A. `<preference for='Magento\Catalog\Block\Product' type='Vendor\Module\Block\Product' />`
- B. `<typename="Magento\Catalog\Block\Product"> q <rewrite class="Vendor\Module\Block\Product" /> </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 the `di.xml` file 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 27

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 to `System > Permissions > Roles` and creating a new role. In the `Resources` section, 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 28

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 29

Which action, if any, can be taken to change the URL key of the product?

- A. The product URL key is generated automatically, so it cannot be changed.
- B. Use URL rewrite to map product id with the custom URL key.
- C. In the product admin form, under the Search Engine Optimization fieldset, the URL key can be set

Answer: C

Explanation:

The URL key of a product is the text that is used to generate the product's URL. This text can be set by the merchant in the product admin form.

The URL key of a product in Adobe Commerce can be changed in the product admin form under the "Search Engine Optimization" fieldset. Here, the URL key can be set or edited manually, allowing for customization beyond the automatically generated value. This field is typically used to ensure that the product URL is search-engine friendly and relevant to the product.

NEW QUESTION 31

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 32

An Adobe Commerce developer is tasked with creating a custom block that will be displayed on every page in the footer of the site. After completing and optimizing the development, the developer notices that the block takes too much time to be generated on each page and decides to store it in the system cache after enabling it for all cache groups. What would be the minimum requirement to achieve this?

- A. Set a value for the cache_Lifetime data property of the block.
- B. Set a value for cache_key data property of the block.
- C. Set values for both cache_lifetime and cache_key data properties of the block.

Answer: A

NEW QUESTION 36

A developer is tasked with creating a new feature in an Adobe Commerce Cloud project. The developer decides to create an integration environment for a better development process. Which Cloud CLI for Commerce command would the developer use?

- A. magento-cloud environment:branch <environment-name> <parent-environment-ID>
- B. magento-cloud create:environment-branch <environment-name> <parent-environment-ID>
- C. magento-cloud environment:create:branch environment-name > <parent-environment-ID>

Answer: A

Explanation:

The Cloud CLI for Commerce command that a developer would use to create an integration environment for a better development process is magento-cloud environment:branch <environment-name> <parent-environment-ID>. This command creates a new branch in the Git repository and a new environment in the Cloud project, using the specified parent environment as a base. The new environment inherits the code, data, and media files from the parent environment. To create a new integration environment for development in an Adobe Commerce Cloud project, the developer would use the Cloud CLI for Commerce command:magento-cloud environment:branch <environment-name> <parent-environment-ID>. This command creates a new environment by branching from the specified parent environment, providing a separate environment for developing new features or testing without affecting the live site.

NEW QUESTION 37

Which index mode is valid?

- A. Update on refresh
- B. Update on invalidate
- C. Update on save

Answer: C

Explanation:

"Update on save" is a valid index mode in Magento, where the index is set to update automatically whenever a change is saved. This mode ensures that the index remains up-to-date with the latest data changes, such as product or category updates, immediately reflecting these changes on the storefront. This real-time indexing is crucial for maintaining data accuracy and consistency across the Magento site, especially in dynamic environments with frequent updates.

NEW QUESTION 40

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 a detailed log of all actions taken on the project, including deployments, enabling developers to track changes and troubleshoot issues that may arise.

NEW QUESTION 42

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:

The catalog_product_view_type_downloadable layout 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 handle `catalog_product_view_type_downloadable` is used. This layout handle allows developers to target downloadable products specifically and apply custom layouts or templates, making option A correct.

NEW QUESTION 47

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="elementTpl" 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="elementTpl" 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="elementTpl" 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="elementTpl" 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 the `elementTpl` to the custom template path within the `arguments` node under the `config` node. 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 48

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 49

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 52

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_buffer_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 53

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.yaml file.* 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 58

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 is `magento-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 63

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 68

How should a grid or form be included in an admin page layout using the UI Component?

- A. `<referenceContainername='content'> q <uiComponentname="example_listing.xml7"></referenceContainer>`
- B. `<referenceContainername='contentf'> 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 and functionality of the UI component, such as grids or forms. This configuration file is located under `view/adminhtml/ui_component/directory` of the corresponding module.

NEW QUESTION 71

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 is `bin/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 73

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

- A. Locally from the command line `bin/magento config:set --lock-config dev/css/minify_files 1` Commit the `app/etc/config.php` file and redeploy.
- B. Update the stores > setting > configuration > Advanced > Developer > css configuration in the Admin Panel.
- C. SSH to the Adobe Commerce Staging 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 on Using ece-tools

NEW QUESTION 76

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 77

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

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

- A. Magento does not allow POST requests to a frontend controller, therefore, the submission functionality will need to be rewritten as an API endpoint.
- B. The developer forgot to implement a `validatePostDataQ` method in their action.
- 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 `form_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 78

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 82

The value of a product attribute in the Adobe Commerce system needs to have a different format before it is displayed. Which attribute model class is responsible for this?

- A. Frontend
- B. Backend
- C. Source

Answer: A

Explanation:

The frontend attribute model class is responsible for formatting the value of a product attribute before it is displayed on the storefront. This class can implement methods such as `getValue`, `getLabel`, `getInputType`, and `getOptionText` to modify the attribute value. The backend attribute model class is responsible for saving and loading the attribute value to and from the database. The source attribute model class is responsible for providing the list of options for an attribute.

Verified References: [Adobe Commerce Developer Guide - Attribute models]

In Adobe Commerce (Magento 2), the formatting of product attribute values before they are displayed is handled by the attribute's frontend model. The frontend model, typically extending `\Magento\Eav\Model\Entity\Attribute\Frontend\AbstractFrontend`, is responsible for preparing the attribute value for display in the frontend, including formatting dates, adding pricing structures, and more. The backend model deals with saving and loading attribute data, while the source model provides options for select attributes.

NEW QUESTION 84

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 in 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 on Email Configuration](#)

NEW QUESTION 88

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.xml` 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 90

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

The current module version is 1.5.4.

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

- A. Implement `Patchversioninterface` and return 1.5.4 on the `getVersion()` method.
- B. Inside `apply()` method, check for module version and run the code if version is less than 1.5.4.
- C. This is not possible.
- D. A module cannot implement both data patch and install scripts.

Answer: A

Explanation:

According to the [Develop data and schema patches guide for Magento 2 developers](#), data patches are classes that contain data modification instructions. They are

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

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

NEW QUESTION 94

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 97

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 \$actionist 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 the catalog route in Magento_Catalog:

? Create a directory structure `app/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 99

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:

The `input_type` attribute of a product EAV attribute specifies the type of input field that will be used to enter the value of the attribute in the admin panel.

The `textarea` input type is used for WYSIWYG fields. If the `input_type` attribute is not set to `textarea`, then the attribute will not appear as WYSIWYG in the admin panel.

To fix this, the developer should set the `input_type` attribute to `textarea`.

NEW QUESTION 102

How can a custom CMS Page be set as a store home page?

- A. In the CMS Page admin grid, select the checkbox for the page under the "Home Page" column.
- B. In the CMS Page admin form, set the "Default Home Page" value to "yes"
- C. In the store configuration, set a custom CMS Page to be a CMS home page

Answer: C

Explanation:

To set a custom CMS Page as a store home page, the developer or merchant should follow these steps:

? In the Admin panel, go to Content > Pages and create or edit a CMS Page that will be used as a home page.

? In the Admin panel, go to Stores > Configuration > General > Web > Default Pages.

? In the CMS Home Page field, select the CMS Page that was created or edited in step 1.

? Save the configuration.

There is no ??Home Page?? column in the CMS Page admin grid or ??Default Home Page?? value in the CMS Page admin form.

Verified References: [Adobe Commerce User Guide - Set up your home page]

In Adobe Commerce, to set a custom CMS page as the store's homepage, you need to go to the store configuration. Specifically, navigate to Content > Design > Configuration, select the relevant store view, and then under the "Default Pages" tab, set the "CMS Home Page" option to your custom CMS page. Options A and B do not exist in the Adobe Commerce admin panel for setting a home page.

NEW QUESTION 107

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 <preference> 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.graphqls` 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.xml` 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 `CartItemPrices` GraphQL schema to include a custom `base_price` field by adding the following code to the module's `etc/schema.graphqls` file:

```
extend type CartItemPrices { base_price: Money! @doc(description: ??The base price of the cart item??) }
```

This code adds a new field called `base_price` to the `CartItemPrices` type and specifies that it is of type `Money` and it is not nullable. The `@doc` directive adds a description for the field that will be shown in the schema documentation. The developer also needs to create a custom resolver class for the `base_price` field and declare it in the `di.xml` file of the module. Verified References: [Magento 2.4 DevDocs] [Magento Stack Exchange]

NEW QUESTION 110

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 store's language. Why does this occur?

- A. `startEnvironmentEmulation()` sets and locks the locale by using the `setLocale()` optional second \$lock parameter, i.
- B. `setLocale($newLocaleCode,true)`, to override and lock the locale of the emulated store.
- C. If this is set and locked initially then the environment emulation will not be able to override this.
- D. `startEnvironmentEmulation()` resets the translation locale to the one of the emulated stores, which overrides the locale the developer has set when the order of `setLocale` and `startEnvironmentEmulation` is used as displayed above.
- E. `setLocale()` does not change translation locale after it has been initially set, the `$this->_translate->emulate($newLocaleCode)` method exists to temporarily modify this by pushing the new locale to the top of the current emulatedLocales stack.

Answer: B

Explanation:

The `startEnvironmentEmulation()` method resets the translation locale to the one of the emulated stores, which overrides the locale the developer has set when the `setLocale()` and `startEnvironmentEmulation()` is used as displayed above.

The correct way to achieve the desired result is to use the `emulate()` method to temporarily modify the translation locale. The following code shows how to do this:

```
PHP
$this->_translate->emulate('en_US');
// Render the template
$this->_translate->revert();
```

This code will set the translation locale to English before rendering the template, and then revert the locale back to the default value after the template has been rendered. The `startEnvironmentEmulation()` method is used to emulate a different store view or website. This can be useful for testing purposes, or for developing features that need to work in different environments.

The `emulate()` method is used to temporarily modify the translation locale. This can be useful for rendering templates in a specific language, or for testing features that need to work in different languages.

NEW QUESTION 111

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 112

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 113

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 `ENHANCEDINTEGRATION`
- 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 on Enhanced Integration Environments

NEW QUESTION 118

.....

Thank You for Trying Our Product

We offer two products:

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

2nd - Questions 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 First Try
- * AD0-E724 Practice Test Questions in Multiple Choice Formats and Updates for 1 Year

100% Actual & Verified — Instant Download, Please Click
[Order The AD0-E724 Practice Test Here](#)