

Adobe.AD0-E722.v2024-08-15.q18

Exam Code:	AD0-E722
Exam Name:	Adobe Commerce Architect Master
Certification Provider:	Adobe
Free Question Number:	18
Version:	v2024-08-15
# of views:	234
# of Questions views:	180
https://www.freeqas.com/qa/Adobe/AD0-E722/Adobe.AD0-E722.v2024-08-15.q18.html	

NEW QUESTION: 1

An Adobe Commerce Architect designs and implements functionality that introduces a new Complex Product Type to the existing Adobe Commerce website. Besides visual demonstration of the new product type, the changes include adjustments to the price index.

The website utilizes a multi-dimensional indexer feature to store the price index. The Architect decides to cover it with integration tests. After creating and running one test, the Architect discovers that database storage is not being fully cleaned.

The test method has the following annotation declaration:

Which adjustment should the Architect make to fix this issue?

- A. Add annotation `@magentoAppIsolation` enabled to method `PHPDoc`
- B. Modify method `PHPDoc` and change annotation `@magentoDbIsolation` to enabled
- C. Create `Customer_ProductType: :Test/_files/{fixture_name}_rollback.php` for every fixture

Answer: ([SHOW ANSWER](#))

Explanation

The issue here is that the database storage is not being fully cleaned after the test is run. The solution is to modify the method `PHPDoc` and change the annotation `@magentoDbIsolation` to enabled. This will ensure that the database storage is fully cleaned after the test is run.

References:

<https://developer.adobe.com/commerce/testing/guide/integration/#database-isolation1>

NEW QUESTION: 2

An Adobe Commerce Architect is reviewing API-functional test code. Some tests send errors to indicate that the customer address does not exist. The test codes show the following:

Which step should the Architect take to fix the test errors?

- A. Change the annotation to `Use@magentoApiDataFixture Magento/Customer/_files/` instead Or `dmagentoDataFixture Magento/Customer/_files/customer_one_address.php`

B. Set the annotation to Use MagentoPersistDataFixture Magento/Customer/_files/customer_one_address.php instead Of @magentoDataFixture Magento/Customer/_files/customer_one_address.php

C. Update the annotation to Specify addressId MagentoDataFixture Magento/Customer/_files/customer_one_address.php with: {Maddress_id: "\$address.id\$"}

Answer: B (LEAVE A REPLY)

Explanation

The issue is being caused by the use of @magentoDataFixture annotation, which creates a temporary data fixture that is rolled back after each test execution¹. This means that the customer address created by the fixture is not persisted in the database and cannot be retrieved by subsequent tests. To fix this, the Architect should use @magentoPersistDataFixture annotation, which creates a permanent data fixture that is not rolled back after each test execution². This way, the customer address created by the fixture will be persisted in the database and can be accessed by subsequent tests. Changing the annotation to use @magentoApiDataFixture or specifying address_id in the annotation will not solve the issue, as they are not related to the persistence of the data fixture³. References: Data fixtures, Persistent data fixtures, API-functional tests

NEW QUESTION: 3

Due to a marketing campaign, a website is experiencing a very large number of simultaneously placed orders, which is affecting checkout performance. The website is in the production deploy mode.

Which two website settings can an Architect optimize to decrease the impact on checkout performance?

(Choose two.)

A. Asynchronous indexing admin panel Setting (Stores > Settings > Configuration > Advanced> Developer

> Grid Settings > Asynchronous indexing) can be enabled by executing the following CLI Command:

bin/Magento config:set dev/grid/async_indexing 1

B. Asynchronous email notifications admin panel setting (stores > settings > configuration > sales > sales Emails > General settings > Asynchronous) can be enabled

C. A new database can be created and the Split Database feature can be automatically configured with the following command: bin/Magento setup:db-schema:split-sales - host"<checkout db host or ip>-

--dbname""<name>" --username"<checkout db username>" --password""<password>"

D. The website deploy mode can be set to siege by executing the following CLI command:

bin/Magento deploy:mode:set siege, provided that it will be changed back to production as soon as the number of simultaneously placed orders decreases to acceptable levels

E. Multithreaded checkout processing admin panel setting (stores > settings > configuration > sales > checkout > General settings > Asynchronous) can be set to a higher value representing the number of PHP threads used exclusively for checkout

Answer: A,C ([LEAVE A REPLY](#))

Explanation

Option A is correct because enabling asynchronous indexing can improve the checkout performance by reducing the database load and avoiding locking issues. Asynchronous indexing allows the indexers to run in the background without affecting the frontend operations. The command `bin/magento config:set dev/grid/async_indexing 1` can be used to enable this option in the production mode¹.

Option C is correct because creating a new database and splitting the sales tables can also improve the checkout performance by distributing the database load and avoiding contention. Splitting the database allows the checkout and order management operations to use a separate master database from the rest of the Magento application tables. The command `bin/magento setup:db-schema:split-sales`

`--host="<checkout db host or ip>" --dbname="<name>" --username="<checkout db username>" --password="<password>"` can be used to configure this feature².

Option B is incorrect because enabling asynchronous email notifications does not affect the checkout performance directly. Asynchronous email notifications allow the order confirmation emails to be sent in batches by a cron job instead of immediately after placing an order. This option can reduce the server load and improve the customer experience, but it does not impact the checkout process itself³.

Option D is incorrect because there is no such deploy mode as siege in Magento 2. The available deploy modes are default, developer, and production. Changing the deploy mode can affect the performance, caching, and error handling of the Magento application, but it does not directly affect the checkout performance⁴.

Option E is incorrect because there is no such admin panel setting as multithreaded checkout processing in Magento 2. The number of PHP threads used for checkout is determined by the web server configuration and the PHP-FPM settings, not by the Magento application settings. Increasing the number of PHP threads may improve the checkout performance, but it also requires more server resources and may cause other issues⁵.

References:

1: Asynchronous indexing | Adobe Commerce Developer Guide

2: Split database performance solution | Adobe Commerce Developer Guide

3: Sales Emails | Adobe Commerce User Guide

4: Set up Magento modes | Adobe Commerce Developer Guide

5: PHP-FPM configuration settings | Adobe Commerce Developer Guide

NEW QUESTION: 4

A custom cron job has been added to an Adobe Commerce system to collect data for several reports. Its crontab.xml configuration is as follows:

The job is data intensive and runs for between 20 and 30 minutes each night.

Within a few days of deployment, it is noticed that the sites sitemap.xml file has not been updated since the new job was added.

What should be done to fix this issue?

A. Change the schedule of the site.generate_cron job to 30 0 * * * so that it runs after the gather_reporting_data job has completed.

B. Create a new cron group for the reporting job, specifying `<use_separate_process>i</use_separate_process>`

C. Break the data gathering job into a number of smaller jobs, so that each individual job runs for a maximum of 5 minutes

Answer: B ([LEAVE A REPLY](#))

Explanation

The issue here is that the gather_reporting_data job is running for a long time and blocking the sitemap_generate job from running. The solution is to create a new cron group for the reporting job and specify `<use_separate_process>i</use_separate_process>` so that the reporting job runs in a separate process and does not block the sitemap_generate job. References:

<https://experienceleague.adobe.com/docs/commerce-cloud-service/user-guide/architecture/starter-architecture.htm>

NEW QUESTION: 5

An Adobe Commerce Architect is planning to create a new action that will add gift registry items to the customer's quote. What should the Architect do to guarantee that private content blocks are updated?

A. Mark the controller by setting no-cache HTTP headers

B. Invalidate the status of gift registry indexers

C. Specify a new action in a sections.xml configuration file

Answer: C ([LEAVE A REPLY](#))

Explanation

Private content blocks are sections of the page that are specific to each customer and are not cached by the server. To update these blocks when a customer performs an action, such as adding a gift registry item to the quote, the Adobe Commerce Architect needs to specify the new action in a sections.xml configuration file.

This file defines which blocks need to be updated for each action and how often they should be updated. By doing this, the Architect can ensure that the private content blocks are refreshed with the latest data from the server. References:

Private content | Magento 2 Developer Documentation

Configure private content | Magento 2 Developer Documentation

NEW QUESTION: 6

A merchant is utilizing an out-of-the-box Adobe Commerce application and asks to add a new reward card functionality for customers. During the code review, the Adobe Commerce Architect

notices the reward_card_number attribute setup created for this functionality is causing the customer attribute to be unavailable in the My account/My rewards page template.

What should be added to set the customer attribute correctly?

- A. group property should be added with a value of 1
- B. system property should be added with a value of true
- C. scope property should be added with a value of global

Answer: A (LEAVE A REPLY)

Explanation

The group property determines which section of the customer account the attribute belongs to. By setting the group property to 1, the reward_card_number attribute will be assigned to the default group, which is the Account Information section. This will make the attribute available in the My account/My rewards page template. References:

<https://experienceleague.adobe.com/docs/commerce-admin/customers/customer-accounts/attributes/attribute-pro>

NEW QUESTION: 7

An Adobe Commerce Architect is working on a scanner that will pull prices from multiple external product feeds. The Architect has a list of vendors and decides to create new config file marketplace.feeds.xml.

Which three steps can the Architect take to ensure validation of the configuration files with unique validation rules for the individual and merged files? (Choose three.)

- A. Implement validation rules in the Converter class for the Config Reader
- B. Create validation rules in marketplace.schema.xsd.
- C. Provide schema to validate a merged file.
- D. Add the Uniform Resource Name to the XSD file in the config XML file.
- E. Provide schema to validate an individual file.
- F. Create a class that implements \Magento\Framework\Config\Datainterface.

Answer: B,C,E (LEAVE A REPLY)

Explanation

The Architect can take the following steps to ensure validation of the configuration files with unique validation rules for the individual and merged files:

Create validation rules in marketplace.schema.xsd. This file defines the structure and constraints of the XML elements and attributes for the marketplace.feeds.xml configuration file. The Architect can use this file to specify the required and optional elements, data types, values, and patterns for the configuration file.

Provide schema to validate a merged file. This schema is used to validate the final configuration file that is generated after merging all the individual configuration files from different modules. The Architect can use this schema to check the consistency and completeness of the merged configuration file.

Provide schema to validate an individual file. This schema is used to validate each individual configuration file from each module before merging them. The Architect can use this schema to check the syntax and validity of each configuration file.

References:

<https://experienceleague.adobe.com/docs/commerce-cloud-service/user-guide/architecture/starter-architecture.htm>

NEW QUESTION: 8

An Adobe Commerce Architect needs to ensure zero downtime during the deployment process of Adobe Commerce on-premises. Which two steps should the Architect follow? (Choose two.)

- A. Enable Config flag Under deployment/blue_green/enabled
- B. Run bin/magento setup:upgrade --dry-run=true to upgrade database
- C. Run bin/magento setup:upgrade - -keep-generated to Upgrade database
- D. Run bin/magento setup:upgrad --convert-old-scripts-true to Upgrade database
- E. Enable Config flag Under developer/zero_down_time/enabled

Answer: A,C (LEAVE A REPLY)

Explanation

Option A is correct because enabling the config flag under deployment/blue_green/enabled is one of the steps to ensure zero downtime during the deployment process of Magento 2 on-premises. This flag enables the blue-green deployment feature, which allows deploying a new version of the Magento application to a separate environment (blue) without affecting the current live environment (green). Once the new version is ready, the traffic can be switched from green to blue with minimal or no downtime¹.

Option C is correct because running bin/magento setup:upgrade --keep-generated is another step to ensure zero downtime during the deployment process of Magento 2 on-premises. This command updates the database schema and data without deleting the generated code and static view files. This way, the Magento application can still serve requests from the cache while the database is being upgraded².

Option B is incorrect because running bin/magento setup:upgrade --dry-run=true does not upgrade the database, but only checks if there are any errors or conflicts in the database schema or data. This command can be used for testing purposes, but it does not affect the deployment process or the downtime³.

Option D is incorrect because there is no such option as --convert-old-scripts-true for the bin/magento setup:upgrade command. This option does not exist in Magento 2 and does not have any effect on the deployment process or the downtime.

Option E is incorrect because there is no such config flag as developer/zero_down_time/enabled in Magento 2. This flag does not exist in Magento 2 and does not have any effect on the deployment process or the downtime.

References:

1: Blue-green deployment | Adobe Commerce Developer Guide

2: Deploy Magento to production | Adobe Commerce Developer Guide

NEW QUESTION: 9

An Adobe Commerce Architect designs a data flow that contains a new product type with its own custom pricing logic to meet a merchant requirement. Which three steps are required when adding a product type with custom pricing? (Choose three.)

- A. Content of the etc/product_types.xml file
- B. Data patch to register the new product type
- C. Hydrator for attributes belonging to the new product type
- D. New price model extending \Magento\Catalog\Model\Product\Type\Price
- E. Custom type model extended from the abstract Product Type model
- F. A new class with custom pricing logic, extending the abstract Product model class

Answer: A,D,E ([LEAVE A REPLY](#))

Explanation

To add a product type with custom pricing, the Architect needs to do the following steps:

Create a content of the etc/product_types.xml file that defines the new product type, its label, model, index priority, and price model. This file is used to register the new product type and its associated classes in Magento1.

Create a new price model that extends \Magento\Catalog\Model\Product\Type\Price and implements the custom pricing logic for the new product type. The price model is responsible for calculating the final price of the product based on various factors, such as special price, tier price, catalog price rules, etc2.

Create a custom type model that extends from the abstract Product Type model (\Magento\Catalog\Model\Product\Type\AbstractType) and overrides the methods related to the product type behavior, such as prepareForCart, getAssociatedProducts, etc. The type model defines how the product type interacts with other components, such as quote, order, cart, etc3. References: How to add a new product type in Magento 2? (MageStackDay mystery question 1) - Magento Stack Exchange Magento 2: How to create custom product types - BeIVG Blog Magento 2: How to create custom product types - BeIVG Blog

NEW QUESTION: 10

An Adobe Commerce Architect needs to create a new customer segment condition to enable admins to specify an Average sales amount' condition for certain segments.

The Architect develops the custom condition under

Vendor\Module\Model\Segment\Condition\AverageSalesAmount with all of its requirements:

During testing, the following error appears:

What should the Architect do to fix the problem?

- A.
- B.
- C.

Answer: B ([LEAVE A REPLY](#))

Explanation

The error is caused by the missing class declaration for the custom condition class. According to the Adobe Commerce documentation, to create a custom customer segment condition, the class must extend the

\Magento\CustomerSegment\Model\Condition\AbstractCondition class and implement the \Magento\CustomerSegment\Model\Condition\CombineInterface interface. The class must also declare its name, label, value type, and attribute code properties. Option B is the only option that correctly declares the class with the required properties and inheritance. Option A and Option C are incorrect because they do not extend the AbstractCondition class or implement the CombineInterface interface, and they do not declare the name, label, value type, or attribute code properties.

References:

Create a customer segment condition | Adobe Commerce Developer Guide

AbstractCondition | Adobe Commerce Developer Guide

NEW QUESTION: 11

An Adobe Commerce Architect creates a stopwords for the Italian locale named stopwordsjtJT.csv and changes the stopwords directory to the following:

<magento_root>/app/code/Customvendor/Elasticsearch/etc/stopwords/ What is the correct approach to change the stopwords directory inside the custom module?

- A.** Add stopwords to the stopwordsDirectory and CustomerVendor_Elasticsearch to the stopwordsModule parameter Of the \Magento\Elasticsearch\SearchAdapter\Query\Preprocessor \Stopwords ClfISS Via di.xml
- B.** Add a new ClaSS implementing \Magento\Framework\Setup\Patch\PatchInterface to modify the default Value Of elasticsearch\customer\stopwordspath in core.conf ig_data table.
- C.** Add stopwords to the stopwordsDirectory parameter of the\Hagento\Elasticsearch\Model \Adapter\Document\DirectoryBuilder ClaSS Via stopwords/it.xml and Adobe Commerce will automatically detect the current module.

Answer: A ([LEAVE A REPLY](#))

Explanation

According to the Adobe Commerce documentation, the correct approach to change the stopwords directory inside a custom module is to use dependency injection to override the default values of the stopwordsDirectory and stopwordsModule parameters of the

\Magento\Elasticsearch\SearchAdapter\Query\Preprocessor\Stopwords class. The stopwordsDirectory parameter specifies the relative path of the stopwords directory from the module directory, while the stopwordsModule parameter specifies the name of the module that contains the stopwords directory. By adding these parameters to the di.xml file of the custom module, the Architect can change the location of the stopwords files without modifying the core code or database.

References:

To change the directory from your module

Configure Elasticsearch stopwords

NEW QUESTION: 12

An Architect is investigating a deployment issue with a server that is configured to work under the symlink directory `/var/www/current`, which lead to the latest released version of the application.

The deployment process performs the following steps:

After the last deployment, the merchant reported that the Adobe Commerce Import/Export functionality to export Customer Main File data is not working. The Architect discovered that the export file is not shown in the list of generated files.

Which change to the deployment process should be performed to solve this issue?

- A.** Restart the consumer process during deployment to use the directory with a new application version for export files.
- B.** Execute Command `config:set export/customr/files_directory /var/releases/{release_number}` toSet the new export path.
- C.** Doable Crontab before deployment and re-launch after deployment.

Answer: (SHOW ANSWER)

Explanation

The issue is that the export file is not shown in the list of generated files. This is because the export path is not set correctly. The solution is to execute the command `config:set export/customr/files_directory`

`/var/releases/{release_number}` to set the new export path. This will ensure that the export file is saved in the correct directory and can be accessed from the Admin Panel. References:

<https://experienceleague.adobe.com/docs/commerce-cloud-service/user-guide/develop/deploy/staging-production>

NEW QUESTION: 13

An Adobe Commerce Architect runs the PHP Mess Detector from the command-line interface using the coding standard provided with Adobe Commerce. The following output appears:

The Architect looks at the class and notices that the constructor has 15 parameters. Five of these parameters are scalars configuring the behavior of `MyService`. The class also contains three constants referencing one other class.

How should the Architect fix the code so that it complies with the coding standard rule?

- A.** Modify the code of `MyService` so that the number of different classes and interfaces referenced anywhere inside the class is fewer than 13.
- B.** Consolidate the constants referencing other classes into a string representation.
- C.** Introduce a new class accepting those five scalars and use it in the constructor and the remaining logic of `MyService`.

Answer: C (LEAVE A REPLY)

Explanation

The issue is being caused by the high coupling between objects (CBO) value of the class `MyService`. CBO is a metric that measures the number of classes that are coupled to a given

class, either by method calls, property or parameter references, inheritance, or constants¹. A high CBO value indicates that the class is too tightly coupled with other classes, which makes it more difficult to maintain, test, and reuse². To reduce the CBO value, the Architect should introduce a new class that encapsulates the five scalar parameters that configure the behavior of MyService. This way, the constructor of MyService will only have one parameter of the new class type, instead of five scalar parameters. This will also make the code more readable and maintainable, as the new class can provide methods to access and manipulate the configuration data. The constants referencing other classes should not be consolidated into a string representation, as this would not reduce the CBO value and would make the code less clear and type-safe³. The number of different classes and interfaces referenced anywhere inside the class is not relevant for the CBO metric, as it only counts the classes that are coupled to the given class¹. References: CBO coupling between object, Coupling Between Object classes (CBO), Cohesion and coupling of an object in OO programming

NEW QUESTION: 14

An Adobe Commerce store owner sets up a custom customer attribute "my.attribute".

An Architect needs to display additional content on the home page, which should display only to Customers with "my.attribute" of a certain value and be the same content for all of them. The website is running Full Page Cache.

With simplicity in mind, which two steps should the Architect take to implement these requirements? (Choose two.)

- A. Add a new context value of "my_attribute" to Magento\Framework\App\Http\Context
- B. Create a Customer Segment and use 'my.attribute' in the conditions
- C. Add a custom block and a pHTML template with the content to the cmsjndexjndex.xml layout
- D. Add a dynamic block with the content to the Home Page
- E. Use customer-data JS library to retrieve "my.attribute" value

Answer: A,D (LEAVE A REPLY)

Explanation

To display additional content on the home page based on a custom customer attribute, the Architect needs to do the following steps:

Add a new context value of "my_attribute" to Magento\Framework\App\Http\Context. This will allow the Full Page Cache to generate different versions of the page for customers with different values of

"my.attribute". The context value can be set using a plugin on the Magento\Customer\Model\Context class.

Add a dynamic block with the content to the Home Page. A dynamic block is a type of content block that can be configured to display only to specific customer segments or conditions. The Architect can use the 'my.attribute' in the conditions of the dynamic block and assign it to the Home Page in the Content > Blocks section of the Admin Panel. References:

Private content | Magento 2 Developer Documentation

Dynamic Blocks | Adobe Commerce 2.3 User Guide - Magento

NEW QUESTION: 15

A merchant asks for a new category attribute to allow uploading an additional mobile image against categories. The merchant utilizes the content staging and preview feature in Adobe Commerce and wants to schedule and review changes to this new mobile image field.

A developer creates the attribute via a data patch and adds it to `view/adminhtml/ui_component/category_form.xml`. The attribute appears against the category in the main form, but does not appear in the additional form when scheduled updates are made. To change this attribute when scheduling new category updates, which additional action should the Architect ask the developer to take?

- A.** The attribute must have its `apply_to` field set to "staging" in the data patch file.
- B.** The attribute must have `<item name="allow_staging" xsi:type="boolean">true</item>` set in the `cjt.gopy_for.xmi` file under the attributes config' section.
- C.** The attribute must also be added to `view/adminhtml/ui_component/catalogstaging_category_update_form.xml`.

Answer: C ([LEAVE A REPLY](#))

Explanation

This action is necessary to make the attribute available for content staging and preview.

According to the Adobe Commerce documentation, the `catalogstaging_category_update_form.xml` file defines the fields that are displayed in the Scheduled Changes section of the category form. The file extends the `category_form.xml` file and adds additional fields that are specific to content staging, such as start and end dates, campaign name, description, etc. To include a custom category attribute in the Scheduled Changes section, the attribute must also be declared in the `catalogstaging_category_update_form.xml` file with the same configuration as in the `category_form.xml` file.

References:

[Content staging | Adobe Commerce Developer Guide](#)

[Create a category attribute | Adobe Commerce Developer Guide](#)

NEW QUESTION: 16

A merchant is using a unified website that supports native Adobe Commerce B2B and B2C with a single store view.

The merchant's objective is to display the B2B account features, such as negotiable quotes and credit limits, in the header of the site on every page for logged-in users who belong to a B2B company account.

Each B2B company possesses its unique shared catalog and customer group, while numerous customer groups for non-B2B customers undergo changes. The merchant insists that this association should not be linked to customer groups.

Which two solutions should the Architect recommend for consideration, taking into account public data and caching? (Choose two.)

- A.** Create a Virtual Type that switches the theme when a user is part of a B2B company so the output can be modified accordingly in the alternate theme.
- B.** Create a new HTTP Context variable to allow for separate public content to be cached for users in B2B companies where the output can be modified accordingly.
- C.** Set whether the current user is part of a B2B company in the customer session and use that data directly to modify the output accordingly.
- D.** Create a new custom condition for customer segments that allow for choosing whether a user is part of a B2B company and then use this segment to modify the output accordingly.
- E.** Check if the current user is part of a B2B company within a block class and modify the output accordingly.

Answer: ([SHOW ANSWER](#))

Explanation

Option B is a valid solution because creating a new HTTP Context variable can allow for differentiating the public content cache for users who belong to a B2B company account. The HTTP Context variable can be used to modify the output of the header block accordingly, without affecting the performance or scalability of the site¹ Option D is also a valid solution because creating a new custom condition for customer segments can enable targeting users who are part of a B2B company account. The customer segment can be used to modify the output of the header block accordingly, using layout updates or dynamic blocks. This solution can also leverage the existing customer segment functionality and avoid custom coding² Option A is not a valid solution because switching the theme based on a virtual type can cause performance issues and increase the complexity of the site maintenance. Moreover, switching the theme can affect the entire site appearance, not just the header block³ Option C is not a valid solution because using the customer session data directly to modify the output of the header block can prevent the public content cache from working properly. The customer session data is private and cannot be cached, so this solution can negatively impact the performance and scalability of the site⁴ Option E is not a valid solution because checking if the current user is part of a B2B company within a block class can also prevent the public content cache from working properly. The block class logic is executed on every request, so this solution can negatively impact the performance and scalability of the site⁵ References:

1:

<https://experienceleague.adobe.com/docs/commerce-cloud-service/user-guide/architecture/starter-architecture.htm>

<https://experienceleague.adobe.com/docs/commerce-cloud-service/user-guide/marketing/customer-segments.htm>

<https://experienceleague.adobe.com/docs/commerce-cloud-service/user-guide/design/themes.html?lang=en> 4:

<https://experienceleague.adobe.com/docs/commerce-cloud-service/user-guide/architecture/starter-architecture.htm>

<https://experienceleague.adobe.com/docs/commerce-cloud-service/user-guide/architecture/starter-architecture.htm>

Valid AD0-E722 Dumps shared by PrepPdf.com for Helping Passing AD0-E722 Exam! PrepPdf.com now offer the **newest AD0-E722 exam dumps**, the PrepPdf.com AD0-E722 exam **questions have been updated** and **answers have been corrected** get the **newest** PrepPdf.com AD0-E722 dumps with Test Engine here: <https://www.preppdf.com/Adobe/AD0-E722-prepaway-exam-dumps.html> (52 Q&As Dumps, **40%OFF Special Discount: Exam-Tests**)

NEW QUESTION: 17

While reviewing a newly developed pull request that refactors multiple custom payment methods, the Architect notices multiple classes that depend on `\Magento\Framework\Encryption\EncryptorInterface` to decrypt credentials for sensitive data. The code that is commonly repeated is as follows:

The Architect needs to recommend an optimal solution to avoid redundant dependency and duplicate code among the methods. Which solution should the Architect recommend?

- A.** Create a common config service class `vendor\Payment\Gateway\Config\Config` under `Vendor\Payment` and use it as a parent class for all of the `Vendor\PaymentModule\Gateway\Config\Config` classes and remove `$scopeConfig` and `$encryptor` dependencies
- B.** Replace all `Vendor\PaymentModule\Gateway\Config\Config` classes with `virtualType` of `\Magento\Payment\Gateway\Config\Config` and set `<user_secret_backend_Model="Magento\Config\Model\Config\Backend\Encrypted" />` under `config.xml`
- C.** Add a plugin after the `getValue` method of `$scopeConfig`, remove the `$encryptor` from dependency and use it in the plugin to decrypt the value if the config name is `user.secret`

Answer: B ([LEAVE A REPLY](#))

Explanation

The Architect should recommend replacing all `Vendor\PaymentModule\Gateway\Config\Config` classes with `virtualType` of `\Magento\Payment\Gateway\Config\Config` and setting `<user_secret_backend_Model="Magento\Config\Model\Config\Backend\Encrypted" />` under `config.xml`. This will avoid redundant dependency and duplicate code among the methods. The `virtualType` of `\Magento\Payment\Gateway\Config\Config` will inherit the functionality of the base class and allow the customization of the constructor arguments, such as the `pathPattern` and `valueHandlerPool`. The `backend_Model` attribute of the `user_secret` field will specify that the value of this field should be encrypted and decrypted by the `\Magento\Config\Model\Config\Backend\Encrypted` class, which implements the `\Magento\Framework\App\Config\ValueInterface` interface and uses the `\Magento\Framework\Encryption\EncryptorInterface` internally¹². This way, the payment modules do not need to depend on the `\Magento\Framework\Encryption\EncryptorInterface` or the

\Magento\Framework\App\Config\ScopeConfigInterface directly, and can use the getValue method of the Magento\Payment\Gateway\Config\Config class to get the decrypted value of the user_secret field3.

References:

How to encrypt system configuration fields in Magento 2 - Mageplaza

Magento 2: How to Encrypt/Decrypt System Configuration Fields - Webkul Blog
Magento 2: How to create custom payment method - BeIVG Blog

NEW QUESTION: 18

A developer needs to uninstall two custom modules as well as the database data and schemas.

The developer uses the following command: bin/magento module:uninstall

Vendor_SampleMinimal Vendor_SampleModifyContent When the command is run from CLI, the developer fails to remove the database schema and data defined in the module Uninstall class.

Which three requirements should the Architect recommend be checked to troubleshoot this issue? (Choose three.)

- A. --remove-schema and --remove-data options are specified as arguments for the CLI command
- B. invoked uninstall() method is implemented in the Uninstall class
- C. invoked uninstall() and uninstallschema are defined in the Uninstall class
- D. --remove-data option is specified as an argument for the CLI command
- E. composer.json file is present and defines the module as a composer package
- F. bin/magento maintenance:enable command should be run in CLI before

Answer: B,D,E (LEAVE A REPLY)

Valid AD0-E722 Dumps shared by PrepPdf.com for Helping Passing AD0-E722 Exam!

PrepPdf.com now offer the **newest AD0-E722 exam dumps**, the PrepPdf.com AD0-E722

exam **questions have been updated** and **answers have been corrected** get the **newest**

PrepPdf.com AD0-E722 dumps with Test Engine here: [https://www.preppdf.com/Adobe/AD0-](https://www.preppdf.com/Adobe/AD0-E722-prepaway-exam-dumps.html)

[E722-prepaway-exam-dumps.html](https://www.preppdf.com/Adobe/AD0-E722-prepaway-exam-dumps.html) (52 Q&As Dumps, **40%OFF** Special Discount: **Exam-Tests**)