



Free Questions for [AD0-E718](#) by [certsdeals](#)

Shared by [Allison](#) on [29-01-2024](#)

For More Free Questions and Preparation Resources

[Check the Links on Last Page](#)

Question 1

Question Type: MultipleChoice

Since the last production deployment, customers can not complete checkout. The error logs show the following message multiple times:

```
main.CRITICAL: Report ID: webapi-61b9fe83f0c3e; Message: Infinite loop detected,  
trace for the looping path
```

The Architect finds a deployed feature that should limit delivery for some specific postcodes.

The Architect sees the following code deployed in/webapi_rest \di .xml and etc\frontend\di xml

```
<type name="Magento\Shipping\Model\Rate\Result">
    <plugin name="RestrictDeliveryMethods" type="Vendor.RestrictDeliveryMethods\Plugin\Shipping\LimitRate
</type>
```

LimitRates.php:

```
public function __construct(
    \Magento\Checkout\Model\Session $session,
    ResultProvider $resultProvider
) {
    $this->session = $session;
    $this->resultProvider = $resultProvider;
}

public function afterGetAllRates(\Magento\Shipping\Model\Rate\Result $subject, array $result): array
{
    return $this->resultProvider->getLimitedRates($this->session->getQuote(), $result);
}
```

Which step should the Architect perform to solve the issue?

Options:

- A-** Inject an instance of \Magento\Quote\API\CartRepositoryInterface and receive cart instance via \$this->cartRepository->get(\$this->session->getQuoteId())
- B-** Replace the injected dependency \Magento\Checkout\Model\Session\With\Magento\Framework\Session\SessionManagerInterface
- C-** Change 'after' plugin with 'around' plugin. The issue is being caused by calling the result provider code after the code of the original method.

Answer:

C

Explanation:

The 'after' plugin is not suitable for modifying the arguments or return value of the original method. The 'around' plugin allows the plugin method to wrap around the original method and modify its behavior and output. Reference: <https://devdocs.magento.com/guides/v2.4/extension-dev-guide/plugins.html#around-methods>

Question 2

Question Type: MultipleChoice

A merchant notices that product price changes do not update on the storefront.

The index management page in the Adobe Commerce Admin Panel shows the following:

- * All indexes are set to 'update by schedule'
- * Their status is 'ready'

* There are no items in the backlog

* The indexes were last updated 1 minute ago

A developer verifies that updating and saving product prices adds the relevant product IDs into the catalog_product_price_cl changelog table.

Which two steps should the Architect recommend to the developer to resolve this issue? (Choose two.)

Options:

- A-** Invalidate the catalog_product_price indexer in the Adobe Commerce Admin Panel so that it is fully reindexed next time the cron runs.
- B-** Manually reindex the catalog_product_price index from the Command line:bin\magento indexer:reindex catalog_product_price.
- C-** Make sure that no custom or third-party modules modify the changelog and indexing process.
- D-** Make sure that the version_id for the price indexer in the mview_state table is not higher than the last entry for the same column in the changelog table and re-synchronize.
- E-** Reduce the frequency of the cron job to 5 minutes so the items have more time to process.

Answer:

C, D

Explanation:

To resolve the issue of product price changes not updating on the storefront, you need to take the following steps:

Make sure that no custom or third-party modules modify the changelog and indexing process. Some modules might interfere with the normal functioning of the indexing mechanism and cause data inconsistency or corruption. You can disable any custom or third-party modules that are related to indexing and check if the issue persists.

Make sure that the `version_id` for the price indexer in the `mview_state` table is not higher than the last entry for the same column in the changelog table and re-synchronize. The `version_id` column in the `mview_state` table indicates the current state of each indexer. If this value is higher than the last entry in the changelog table, it means that the indexer has skipped some records and needs to be re-synchronized. You can use the `bin/magento indexer:reset` command to reset the state of the price indexer and then run `bin/magento indexer:reindex` to reindex it.

: <https://devdocs.magento.com/guides/v2.4/extension-dev-guide/indexing.html> : <https://devdocs.magento.com/guides/v2.4/config-guide/cli/config-cli-subcommands-index.html>

Question 3

Question Type: MultipleChoice

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:

```

namespace Vendor\PaymentModule\Gateway\Config;

class Config extends \Magento\Payment\Gateway\Config\Config
{
    ...
    public function __construct(
        ...
        ScopeConfigInterface $scopeConfig,
        EncryptorInterface $encryptor,
        ...
    ) {
        parent::__construct($scopeConfig, $methodCode, $pathPattern);
        $this->scopeConfig = $scopeConfig;
        $this->encryptor = $encryptor;
    }

    public function getUserSecret(): string
    {
        return $this->encryptor->decrypt(
            $this->scopeConfig->getValue('payment/method_code/user_secret')
        );
    }
}

```

In each module, the user_secret config is declared as follows:

```

<field id="user_secret" translate="label" type="obscure" sortOrder="20" showInDefault="1" >
    <label>Secret Key</label>
    <backend_model>Magento\Config\Model\Config\Backend\Encrypted</backend_model>
</field>

```

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

Options:

- A-** Replace all Vendor\PaymentModule\Gateway\Config\Config Classes With virtualTyp- Of Magento\Payxer.t\Gateway\Config\Config and Set <user_secret backend_Model='Magento\Config\Model\Config\Backend\Encrypted' /> under ccnfig.xml
- B-** Add a plugin after the getvalue method of \$sccpeConfig, remove the \$encryptor from dependency and use it in the plugin to decrypt the value if the config name is 'user.secret'?
- C-** 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 Vender \EaymentModule\Gateway\Config\Config Classes and remove \$sccpeConfig and \$encryptor dependencies

Answer:

C

Explanation:

To avoid redundant dependency and duplicate code among the methods, you need to use the following solution:

Create a standard controller route and mapping the internal URLs (such as news/article/view/id/1) to rewrites that are generated on save and then stored in the URL rewrites table. This solution will leverage the built-in URL rewrite functionality of Magento to handle the news requests. You can create a standard controller route that handles the internal URLs and loads the news article by its ID. Then, you can generate URL rewrites for each news article based on its date and URL key and store them in the URL rewrites table. This way, you can avoid using a custom router or a plugin and reduce the code complexity.

: <https://devdocs.magento.com/guides/v2.4/extension-dev-guide/routing.html> : <https://devdocs.magento.com/guides/v2.4/extension-dev-guide/url-rewrite.html>

Question 4

Question Type: MultipleChoice

An Adobe Commerce Architect is working on a sales campaign to present a new product on the site that allows the purchase of a pre-defined set of products with a discount. Each product in the set should have a separate stock and tax class.

One requirement is to use a third-party system to build reports with REST API to fetch the following data:

- * SKU
- * Qty
- * Original price
- * Sales price
- * Tax amount

Which solution should the Architect use to meet these requirements?

Options:

A- * Create Fixed Bundle Product for gathering simple products;

* Manage price for every selected option;

* Add extension attribute `original_simple_price` for `\Magento\Sales\Api\Data\OrderItemExtensionInterface` and populate value with price of simple product;

B- * Create Dynamic Bundle Product for gathering simple products;

* Utilize Content Staging to manage special prices for bundle products on time for the campaign;

* Expose required data via Adobe Commerce Order API;

C- * Create Grouped Product and Create after plugin on

`\Magento\GroupedProduct\Model\Product\Type\Grouped:preparedForCarrAdvanced` for bunch products ordering;

* Utilize Content Staging to manage special prices on time for the campaign for simple products;

* Expose required data via Adobe Commerce Order API;

Answer:

A

Explanation:

To meet the requirements, the Architect should use the following solution: * Create Fixed Bundle Product for gathering simple products. This will allow the purchase of a pre-defined set of products with a discount and separate stock and tax class for each product. * Manage price for every selected option. This will allow setting the original and sales price for each product in the bundle. * Add extension attribute `original_simple_price` for `\Magento\Sales\Api\Data\OrderItemExtensionInterface` and populate value with price of simple product. This

will allow exposing the original price of the simple product via REST API along with other required data

a. Option B is incorrect because Dynamic Bundle Product will not allow setting a pre-defined set of products with a discount. Option C is incorrect because Grouped Product will not allow setting a discount for the whole set of products. Reference:

<https://docs.magento.com/user-guide/catalog/product-create-bundle.html> https://devdocs.magento.com/guides/v2.4/extension-dev-guide/extension_attributes/adding-attributes.html

Question 5

Question Type: MultipleChoice

An Adobe Commerce Architect needs to scope a bespoke news section for a merchant's Adobe Commerce storefront. The merchant's SEO agency requests that the following URL structure:

news/{date}/{article_url_key} where {date} is the publication date of the article, and {article_url_key} is the URL key of the article.

The Architect scopes that a news entity type will be created. The date and URL key data will be stored against each record and autogenerated on save. The values will be able to be manually overridden.

The Architect needs to manage routing this functionality and adhere to best practice.

Which two options should the Architect consider to meet these requirements? (Choose two.)

Options:

- A-** Create a standard controller route and an Index/Index index controller class that loads the relevant news article by matching the URL date and URL key parts.
- B-** Create an observer that listens to the controllers_front_send_response_before event, looks for the mm portion of the URL, and If it matches, loads the relevant news article by matching the URL date and URL key parts.
- C-** Create a plugin that intercepts lu^jentoXFraBeworkUppXActien::executed, looks for the news portion of the URL and if it matches, loads the relevant news article by matching the URL date and URL key parts.
- D-** Create a standard controller route and mapping the internal URLs (such as news/article/view/id/1) to rewrites that are generated on save and then stored in the URL rewrites table.
- E-** Create a custom router that runs before the standard router and matches the news portion of the URL. then looks for and loads a news article by matching the date and URL key parts of the URL.

Answer:

A, E

Explanation:

To manage routing this functionality and adhere to best practice, you need to consider the following options:

Create a standard controller route and an Index/Index index controller class that loads the relevant news article by matching the URL date and URL key parts. This option will create a simple and straightforward way to handle the news requests using the standard

Magento routing mechanism. The Index/Index controller class will receive the date and URL key parameters from the request and use them to load the news article model from the database.

Create a custom router that runs before the standard router and matches the news portion of the URL, then looks for and loads a news article by matching the date and URL key parts of the URL. This option will create a more flexible and customizable way to handle the news requests using a custom router class that implements \Magento\Framework\App\RouterInterface. The custom router class will check if the request path starts with news, then extract the date and URL key parts from the path and use them to load the news article model from the database.

: <https://belvg.com/blog/how-to-create-custom-router-in-magento-2.html> : <https://devdocs.magento.com/guides/v2.4/extension-dev-guide/routing.html>

Question 6

Question Type: MultipleChoice

An existing Adobe Commerce website is moving to a headless implementation.

The existing website features an "All Brands" page, as well as individual pages for each brand. All brand-related pages are cached in Varnish using tags in the same manner as products and categories.

Two new GraphQL queries have been created to make this information available to the frontend for the new headless implementation:

```
type Query {
  brands (
    search: String @doc(description: "Search against brand names (partial matches)")
    pageSize: Int = 20 @doc(description: "Number of brand results to return")
    currentPage: Int = 1 @doc(description: "Page number to return")
  ) : BrandsResult
  @resolver(class: "ClientName\\ProductBrandsGraphQL\\Model\\Resolver\\Brands")
  brand (
    urlKey: String! @doc(description: "Match against brand url key")
  ) : Brand
  @resolver(class: "ClientName\\ProductBrandsGraphQL\\Model\\Resolver\\BrandByUrlKey")
}
```

During testing, the queries sometimes return out-of-date information.

How should this problem be solved while maintaining performance?

Options:

- A-** Each GraphQL query's resolver class should inject MagentoGraphQLCacheModelCacheableQuery and call setCacheValidity (true) on it as part of the resolver's resolve function
- B-** Specify a @cache (cacheable: false directive for each GraphQL query, making sure that the data returned is not cached, and is up to date
- C-** Specify a @cache (cacheIdentity: path\\T\\identityClass) directive for each GraphQL query, corresponding to a class that adds cache tags for relevant brands and associated products

Answer:

C

Explanation:

This option would allow caching GraphQL query results using Varnish or Fastly with proper invalidation and differentiation mechanisms.

To solve the problem of out-of-date information in GraphQL queries, the Architect should specify a `@cache (cacheIdentity: path\T\identityClass)` directive for each GraphQL query, corresponding to a class that adds cache tags for relevant brands and associated products. This will allow Varnish to invalidate the cache for these queries when the brand or product data changes. Option A is incorrect because it will make the queries cacheable without any cache tags, which will not solve the problem. Option B is incorrect because it will disable caching for these queries entirely, which will degrade performance. Reference: <https://devdocs.magento.com/guides/v2.4/graphql/caching.html>

Question 7

Question Type: MultipleChoice

An Adobe Commerce store owner sets up a custom customer attribute "my.attribute" (type int).

An Architect needs to display customer-specific content on the home page to Customers with "my.attribute" greater than 3. The website is running Full Page Cache.

Using best practices, which two steps should the Architect take to implement these requirements? (Choose two.)

Options:

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

Answer:

A, C

Explanation:

To display customer-specific content on the home page to Customers with "my.attribute" greater than 3, you need to use the following steps:

Use customer-data JS library to retrieve "my.attribute" value. The customer-data JS library allows you to access the customer data stored in the local storage of the browser. You can use the get() method to get the value of "my.attribute" for the current customer.

Add a new context value of "my.attribute" to `Magento\Framework\App\Http\Context`. The HTTP context is a mechanism that allows you to vary the cache of a page based on some parameters. By adding a new context value of "my.attribute", you can cache separate versions of the home page for customers with different values of "my.attribute".

Add a custom block and a phtml template with the content to the `cms_index_index.xml` layout. This step will create a block that will display the customer-specific content on the home page. You can use the phtml template to check the value of "my.attribute" and render the content accordingly.

: <https://devdocs.magento.com/guides/v2.3/extension-dev-guide/cache/page-caching/public-content.html> :

https://devdocs.magento.com/guides/v2.3/howdoi/checkout/checkout_customize.html

To Get Premium Files for AD0-E718 Visit

<https://www.p2pexams.com/products/ad0-e718>

For More Free Questions Visit

<https://www.p2pexams.com/adobe/pdf/ad0-e718>

